

# **METODOLOGIAS PARA EL APRENDIZAJE DE CONVERTIDORES AC-DC**

**JOVANNY BEDOYA GUAPACHA**

**Maestría en Ingeniería Eléctrica**

**Facultad de Ingenierías**

**Universidad Tecnológica de Pereira**

**Pereira, Febrero de 2013**

# **METODOLOGIAS PARA EL APRENDIZAJE DE CONVERTIDORES AC-DC**

**JOVANNY BEDOYA GUAPACHA**

**Tesis o trabajo de investigación presentado como requisito parcial para  
optar al título de:  
Magister en Ingeniería**

**Director (a):  
Msc. Alfonso Alzate Gómez**

**Maestría en Ingeniería Eléctrica**

**Facultad de Ingenierías**

**Universidad Tecnológica de Pereira**

**Pereira, Febrero de 2013**

## **Agradecimientos**

A Dios por haberme permitido llegar hasta este punto y haberme dado salud para lograr mis objetivos.

A mi gran esposa Luz Piedad Valencia Montoya por su incondicional apoyo.

A mi director de tesis, Alfonzo Alzate, por su paciencia, su influencia y confianza en mí.

A mi gran amigo y tutor Hernando Ríos.

A los almacenistas y secretaria que tanto quiero, Mario, Francisco, Gustavo y Clara Inés.

A mi madre Luz Mery Guapacha y mi hermana Adriana, por apoyarme.

A mis suegros Amparo y José Luis.

A mis jurados Andrés German Holguín y Luis Hernando Ríos

A la Universidad Tecnológica de Pereira.

## Contenido

Capítulo 1. Introducción .....	12
1.1 <b>Justificación</b> .....	12
1.2 <b>Objetivos</b> .....	13
1.2.1 Objetivo General: .....	13
1.2.1 Objetivos específicos: .....	13
1.3 <b>Metodología</b> .....	14
Capitulo2. Desarrollo Teórico .....	15
2.1 <b>EL DIODO DE POTENCIA</b> .....	15
2.1.1 Características estáticas .....	16
2.1.2 Modelos estáticos del diodo .....	17
2.1.3 <b>Características dinámicas</b> .....	18
2.1.4 Tiempo de recuperación directo: .....	19
2.2 <b>EL SCR</b> .....	19
2.2.1 CARACTERÍSTICAS GENERALES .....	20
2.2.2 CARACTERÍSTICAS ESTÁTICAS .....	21
2.2.3 ÁREA DE DISPARO SEGURO .....	21
2.2.4 Características Dinámica .....	22
2.2.5 CARACTERÍSTICAS DE CONMUTACIÓN .....	23
2.3 <b>Rectificadores con Diodos o no Controlados</b> .....	24
2.3. 1 Parámetros de rendimiento de la salida de los rectificadores: .....	25
2.3 2 Parámetros de rendimiento respecto a la entrada de los rectificadores	26
2.4 <b>Rectificadores Controlados</b> .....	28
2.5 <b>Principio de operación del convertidor controlado por fase</b> .....	29
2.6 <b>Semiconvertidor monofásico</b> .....	31
2.6.1 Operación: .....	31
2.7 <b>Convertidor Monofásico Completo</b> .....	33
2.7.1 Modo de funcionamiento discontinuo: .....	37
2.7.2 Modo de funcionamiento continuo: .....	37
2.8 <b>Lenguaje de Programación Java</b> .....	39

2.8.1 Orientado a objetos .....	40
2.8.2 Independencia de la plataforma .....	41
2.8.3 El recolector de basura .....	42
2.8.4 Sintaxis .....	43
2.8.5 Applets.....	44
2.8.6 Servlets.....	44
2.8.7 Aplicaciones con ventanas .....	44
2.9 Lenguaje de programación Python .....	45
2.9.1 Características: .....	45
2.10 Arquitectura cliente servidor en JAVA.....	47
2.10.1 ELEMENTOS PRINCIPALES .....	48
2.10.2 ELEMENTOS DE LA ARQUITECTURA CLIENTE/SERVIDOR .....	50
2.10.3 Infraestructura de redes.....	52
2.10.4 TIPOS DE CLIENTES.....	54
2.10.5 TIPOS DE SERVIDOR .....	54
2.10.6 ESTILOS DEL MODELO CLIENTE SERVIDOR.....	55
2.11 Sistema de Procesamiento (Microcontrolador).....	56
2.11.1 EL MICROCONTROLADOR MC9S08QG8 .....	57
Capítulo 3. Diseño y construcción de rectificador monofásico controlado y control de fase monofásico .....	72
3.1 Modulo Rectificador Monofásico no Controlado, Semicontrolado y Controlado.....	72
3.1.1 Control On-Line.....	73
3.2 Modulo Control de fase .....	75
3.3 Sistema de Alimentación del sistema .....	76
3.4 Sistema de detección de cruce por cero .....	76
3.5 Captura y Acondicionamiento de señal de tensión .....	78
3.6 Captura y Acondicionamiento de señal de Corriente .....	81
3.7 Temporizador LM555 .....	84
3.7.1 Operación en modo monoestable.....	85
3.8 Sistema de comunicación RS232 .....	86

3.8.1 Envío de datos .....	88
<b>3.9 Software del módulo no controlado, semicontrolado, controlado. ....</b>	<b>89</b>
3.9.1 Implementación protocolo comunicación serial a 256000 bps en PYTHON.....	89
3.9.2 Implementación de Procesamiento, conectividad (Cliente-Servidor) y toma de decisiones en JAVA .....	93
<b>3.10 Implementación de la comunicación, captura de señales y el procesamiento en los microcontroladores Motorola Freescale. ....</b>	<b>95</b>
3.10.1 Comunicación: .....	95
3.10.2 Captura de señales Análogas y digitales: .....	98
Capitulo 4. Validación de los módulos.....	102
4.1 Control PI al Convertidor Controlado .....	102
4.2 Modulo Control de Fase .....	105
4.3 Modulo Convertidor Controlado, Semicontrolado y no Controlado....	109
Conclusiones.....	117
Bibliografía .....	118

## Tabla de Figuras

Figura 1. Diodo.....	15
Figura 2. Curva característica.....	15
Figura 3 .....	17
Figura 4.Recuuperracion inversa del diodo.....	18
Figura 5.Recuperacion directa del diodo.....	19
Figura 6,Simbolo .....	19
Figura 7.Union.....	20
Figura 8.Curva caractristica.....	20
Figura 9.Característica de puerta del tiristor .....	22
Figura 10.Ángulo de bloqueo y conducción de un tiristor .....	23
Figura 11.Circuito rectificador monofásico .....	24
Figura 12.Parámetros de rendimiento .....	26
Figura 13.Región de operación del convertidor.....	30
Figura 14.Semiconvertidor monofásico .....	32
Figura 15.Convertidor monofásico .....	33
Figura 16. Forma de onda del rectificador monofásico .....	34
Figura 17. Entrada y salida del rectificador controlado.....	35
Figura 18.Tension y Corriente en la carga. ....	37
Figura 19. Arquitectura Java .....	39
Figura 20. Arquitectura Cliente Servidor .....	49
Figura 21. Distribución Cliente Servidor .....	50
Figura 22. Aplicación arquitectura Cliente Servidor.....	51
Figura 23.Modelo Cliente Servidor .....	52
Figura 24. Cliente Flaco .....	54
Figura 25. Cliente Gordo .....	54
Figura 26. MC9S08QG8.....	57
Figura 27. Arquitectura MC9S08QG8 .....	57

Figura 28. Modulo IRQ pin .....	60
Figura 29. Mascara de Interrupcion.....	61
Figura 30. Puertos Keyboard.....	62
Figura 31. Mascara interrupcion KBD .....	63
Figura 32. Señal Analoga .....	64
Figura 33. Muestreo del Conversor ADC .....	65
Figura 34. Puerto y Bloques del Módulo TPM .....	69
Figura 35. Diagrama de Bloques TPM .....	70
Figura 36. Función Input Capture.....	71
Figura 37. A Modulo Rectificador Monofásico no Controlado, Semicontrolado y Controlado.....	72
Figura 38. B Modulo Rectificador Monofásico no Controlado, Semicontrolado y Controlado.....	73
Figura 39. A . Modulo Control de fase.....	75
Figura 40. B . Modulo Control de fase.....	75
Figura 41. Esquema Circuital .....	76
Figura 42. Circuito realizado.....	76
Figura 43. MOC 4N25 .....	77
Figura 44. Detección de cruce por cero .....	77
Figura 45. Señal obtenida .....	77
Figura 46. Montaje físico .....	78
Figura 47. Divisor de tensión.....	78
Figura 48. Amplificador de instrumentación AD202JN .....	79
Figura 49. Diagrama de Bloques.....	79
Figura 50. Circuito de acondicionar la señal para su digitalización .....	80
Figura 51. Señal obtenida .....	81
Figura 52. ACS714.....	81
Figura 53. Diagrama interno del ASC714.....	82
Figura 54. Circuito impreso ACS714 .....	83
Figura 55. Tensión vs Corriente .....	83
Figura 56. Señal obtenida del sensor de corriente .....	83



Figura 57. Estructura interna y diagrama de conexión del LM555 .....	84
Figura 58. LM555 en configuración monoestable.....	85
Figura 59. FT232.....	86
Figura 60. Tarjeta USB-Serial conFT232RL.....	87
Figura 61.Comunicación asincrónica .....	87
Figura 62.Comunicación sincrónica .....	88
Figura 63. Pines de comunicación .....	88
<b>Comunicación serial en los microcontroladores Motorola-Freescale.....</b>	<b>88</b>
Figura 64. Ventana del Aplicativo.....	93
Figura 65. Señales obtenidas de tensión y corriente.....	94
Figura 66. Señal Digital Cruce por cero .....	99
Figura 67. Exitacion a los optoacopladores señal de entrada (rojo), señal de salida (azul). .....	101
Figura 68. Circuito de excitación al tiristor.....	101
Figura 69. Controlador PI .....	102
Figura 70. Planta con excitación escalón .....	103
Figura 71. Planta Control Proporcional .....	104
Figura 72. Planta con control Proporcional Integral.....	104
Figura 73. Control de un semiciclo carga R .....	105
Figura 74. Control a 0° de carga R.....	105
Figura 75. Control a 90° de carga R.....	105
Figura 76. Control de un semiciclo con carga R – L.....	106
Figura 77. Montaje Fisico .....	106
Figura 78. Control a 0° de carga R - L.....	106
Figura 79. Control a 90° de carga R – L.....	107
Figura 80. Control de fase.....	107
Figura 81.Control a 0° de carga R.....	107
Figura 82. Control a 90° de carga R.....	108
Figura 83. Control de fase con carga R - L .....	108
Figura 84. Control a 0° de carga R – L.....	108
Figura 85. Control a 90° de carga R - L.....	109

Figura 86. Rectificador no controlado con carga R .....	109
Figura 87. Convertidor a $0^\circ$ carga R.....	109
Figura 88. Convertidor a $0^\circ$ carga R visto desde consola .....	110
Figura 89. Convertidor a $90^\circ$ carga R.....	110
Figura 90. Convertidor a $90^\circ$ carga R visto desde la consola.....	110
Figura 91. Convertidor AC/DC No Controlado CON carga R – L .....	111
Figura 92. Montaje fisico .....	111
Figura 93. Control a $0^\circ$ con carga R - L.....	111
Figura 94. Convertidor AC/DC Semi-Controlado CON carga R .....	112
Figura 95. Control a $0^\circ$ con carga R.....	112
Figura 96. Control a $90^\circ$ con carga R.....	112
Figura 97. Convertidor AC/DC Semicontrolado CON carga R – L .....	113
Figura 98. Control a $0^\circ$ con carga R – L.....	113
Figura 99. Control a $90^\circ$ con carga R – L.....	113
Figura 100. Convertidor AC/DC Controlado CON carga R.....	114
Figura 101. Control a $0^\circ$ con carga R.....	114
Figura 102. Control a $90^\circ$ con carga R.....	114
Figura 103. Convertidor AC/DC Controlado CON carga R – L.....	115
Figura 104. Montaje fisico .....	115
Figura 105. a. Control a $0^\circ$ con carga R – L.....	115
Figura 106. b. Control a $0^\circ$ con carga R - L.....	116
Figura 107. Control a $90^\circ$ con carga R – L.....	116

## **Resumen**

En este trabajo se presenta el diseño e implementación del rectificador monofásico controlado, semicontrolado y no controlado, y el diseño e implementación de control de fase con sus respectivas variantes, este trabajo tiene como elemento principal el control del sistema on line (Internet, Cliente - Servidor) a través de software libre, por medio de alguna técnica de control digital por ejemplo control Proporcional, Proporcional Integral, PID u otra.

Se desarrollaron etapas de recopilación de Información bibliográfica, seguidamente se pasó a la etapa de diseño en la cual se especificaron parámetros de los componentes requeridos para el circuito, tomando en cuenta las capacidades de los equipos disponibles en el laboratorio y en el mercado, además de la simplicidad requerida para que sea fácil de implementar.

Posteriormente a la etapa de diseño y especificación se realizaron predicciones del comportamiento del circuito utilizando tanto, modelos matemáticos basados en el comportamiento ideal esperado de los, de manera que se tenga un parámetro de referencia para el estudio de desempeño del circuito. Este último consiste en la construcción del circuito y en la captura de datos reales. Culminadas las etapas se llegó a un circuito sencillo con un desempeño que permite que los datos realizados se puedan considerar valiosos para el estudio del comportamiento de un rectificador controlado, semicontrolado y no controlado de onda completa.

## Capítulo 1. Introducción

### 1.1 Justificación

En muchas aplicaciones industriales, la carga eléctrica alimentada requiere una tensión continua. La conversión CA/CC es realizada por convertidores estáticos de energía, comúnmente denominados **rectificadores**. Por tanto, un rectificador es un sistema electrónico de potencia cuya función es convertir una tensión alterna en una tensión continua. En una primera clasificación, podemos diferenciar los rectificadores de acuerdo con el número de fases de la tensión alterna de entrada. Los rectificadores no controlados son aquellos que utilizan diodos como elementos de rectificación, en cuanto que los controlados utilizan tiristores o transistores.

Hoy en día, el uso de circuitos de electrónica de potencia abarca un mercado muy amplio, sus usos van desde control a conmutación y conversión de energía, se encuentran en dispositivos como rectificadores, control de fase, variadores de velocidad, fuentes de alimentación ininterrumpida, fuentes de alimentación conmutadas, entre muchos otros. Dado al amplio campo en el que se desenvuelve esta rama de la ingeniería eléctrica es de esperarse del estudiante un conocimiento en lo que concierne tanto a la teoría como a la puesta en práctica de circuitos de este tipo, es por tanto que se desea realizar una metodología de aprendizaje practico con el propósito de guiar al estudiante a través de las etapas básicas de aplicaciones comunes (a pesar de la antigüedad del esquema básico del circuito) como lo es un rectificador monofásico controlado por medio de SCR.

La planta física de los laboratorios en ocasiones no tienen la capacidad de albergar grupos grandes de estudiantes por lo tanto es indispensable crear mecanismos que también permitan al estudiante realizar prácticas de circuitos como rectificación controlada desde lugares remotos por medio del esquema cliente servidor a través de software libre como el lenguaje Java y Python, que permiten visualizar lo que ocurre en el circuito a controlar.

## 1.2 Objetivos

### 1.2.1 Objetivo General:

Diseñar e implementar un banco de pruebas que permita estudiar experimentalmente el comportamiento de las diferentes topologías de rectificación controlada en diferentes condiciones de conmutación a través de control on line.

### 1.2.2 Objetivos específicos:

- Construcción del banco de pruebas (que incluye los disparo de compuerta, los sistemas de control y la fuente de CD)
- Diseño e implementación de la tarjeta de disparo de los tiristores
- Construir el circuito y validar los resultados usando equipo de fácil disponibilidad para el estudiante, de manera que se acerque a las condiciones de laboratorio en que se va a trabajar.
- Realizar el diseño del circuito.
- Comunicación cliente servidor.
- Realización de algunas pruebas para comprobar el buen funcionamiento del banco de pruebas.
- Ejemplo control PI

### 1.3 Metodología

La metodología usada en el proyecto consiste en una investigación bibliográfica de proyectos realizados anteriormente relacionados con el tema, con el fin de recopilar todos los datos obtenidos anteriormente para facilitar el diseño del circuito deseado. Además se recopilará información de libros sobre ciertos parámetros adicionales y especificaciones del fabricante de componentes esenciales del circuito.

Diseño de las etapas necesarias para el circuito, especificando cada elemento con respecto a las especificaciones de carga, además como factor importante a la hora de diseño se considerarán las capacidades de los equipos disponibles en el mercado de manera que se diseñe un circuito sencillo.

Diseño de la etapa de adquisición de las señales análogas e interpretación de las mismas.

Diseño de la etapa de comunicaciones SCI con interfaz RS232 a USB del módulo de rectificación.

Conexión on line entre cliente servidor

Finalmente se procederá a la puesta en operación del módulo con un circuito semicontrolado o controlado, y la comparación de los parámetros con respecto a los esperados, para poder llegar finalmente a conclusiones acerca de la validez del mismo.

## Capítulo 2. Desarrollo Teórico

### 2.1 EL DIODO DE POTENCIA

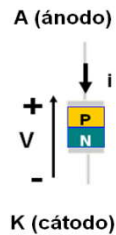


Figura 1. Diodo

Uno de los dispositivos más importantes de los circuitos de potencia son los diodos, aunque tienen, entre otras, las siguientes limitaciones: son dispositivos unidireccionales, no pudiendo circular la corriente en sentido contrario al de conducción. El único procedimiento de control es invertir el voltaje entre ánodo y cátodo. Los diodos de potencia se caracterizan porque en estado de conducción, deben ser capaces de soportar una alta intensidad con una pequeña caída de tensión. En sentido inverso, deben ser capaces de soportar una fuerte tensión negativa de ánodo con una pequeña intensidad de fugas.



Figura 2. Curva característica

El diodo responde a la ecuación:

$$I = I_S \times (e^{qV/KT} - 1)$$

La curva característica será la que se puede ver en la parte superior, donde:

$V_{RRM}$ : tensión inversa máxima

$V_D$ : tensión de codo.

A continuación podemos apreciar las características más importantes del diodo, las cuales podemos agrupar de la siguiente forma:

- Características estáticas:
  - Parámetros en bloqueo (polarización inversa).
  - Parámetros en conducción.
  - Modelo estático.
- Características dinámicas:
  - Tiempo de recuperación inverso ( $t_{rr}$ ).
  - Influencia del  $t_{rr}$  en la conmutación.
  - Tiempo de recuperación directo.
- Potencias:
  - Potencia máxima disipable.
  - Potencia media disipada.
  - Potencia inversa de pico repetitivo.
  - Potencia inversa de pico no repetitivo.
- Características térmicas.
- Protección contra sobre intensidades.

### 2.1.1 Características estáticas

Parámetros en bloqueo

- Tensión inversa de pico de trabajo ( $V_{RWM}$ ): es la que puede ser soportada por el dispositivo de forma continuada, sin peligro de entrar en ruptura por avalancha.
- Tensión inversa de pico repetitivo ( $V_{RRM}$ ): es la que puede ser soportada en picos de 1 ms, repetidos cada 10 ms de forma continuada.
- Tensión inversa de pico no repetitiva ( $V_{RSM}$ ): es aquella que puede ser soportada una sola vez durante 10ms cada 10 minutos o más.



- Tensión de ruptura ( $V_{BR}$ ): si se alcanza, aunque sea una sola vez, durante 10 ms el diodo puede destruirse o degradar las características del mismo.
- Tensión inversa continua ( $V_R$ ): es la tensión continua que soporta el diodo en estado de bloqueo.

#### Parámetros en conducción

- Intensidad media nominal ( $I_{F(AV)}$ ): es el valor medio de la máxima intensidad de impulsos sinusoidales de  $180^\circ$  que el diodo puede soportar.
- Intensidad de pico repetitivo ( $I_{FRM}$ ): es aquella que puede ser soportada cada 20 ms, con una duración de pico a 1 ms, a una determinada temperatura de la cápsula (normalmente  $25^\circ$ ).
- Intensidad directa de pico no repetitiva ( $I_{FSM}$ ): es el máximo pico de intensidad aplicable, una vez cada 10 minutos, con una duración de 10 ms.
- Intensidad directa ( $I_F$ ): es la corriente que circula por el diodo cuando se encuentra en el estado de conducción.

#### 2.1.2 Modelos estáticos del diodo



Figura 3 modelo ideal y estático del diodo

Los distintos modelos del diodo en su región directa (modelos estáticos) se representan en la figura superior. Estos modelos facilitan los cálculos a realizar, para lo cual debemos escoger el modelo adecuado según el nivel de precisión que necesitemos.

### 2.1.3 Características dinámicas

#### Tiempo de recuperación inverso

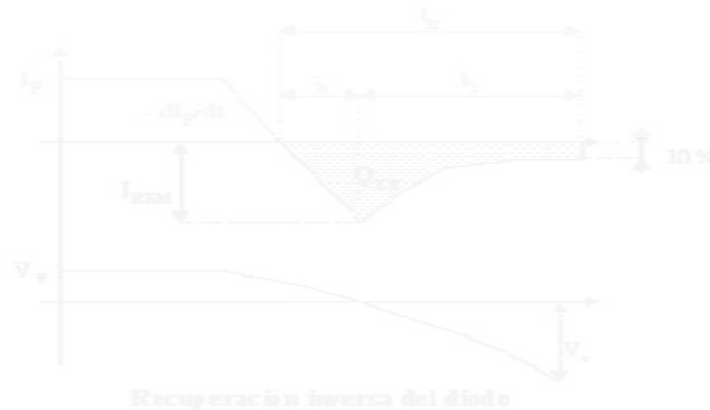


Figura 4. Recuperracion inversa del diodo

El paso del estado de conducción al de bloqueo en el diodo no se efectúa instantáneamente.

$t_a$  (tiempo de almacenamiento): es el tiempo que transcurre desde el paso por cero de la intensidad hasta llegar al pico negativo.

$t_b$  (tiempo de caída): es el tiempo transcurrido desde el pico negativo de intensidad hasta que ésta se anula, y es debido a la descarga de la capacidad de la unión polarizada en inverso. En la práctica se suele medir desde el valor de pico negativo de la intensidad hasta el 10 % de éste.

$t_{rr}$  (tiempo de recuperación inversa): es la suma de  $t_a$  y  $t_b$ .

$Q_{rr}$ : se define como la carga eléctrica desplazada, y representa el área negativa de la característica de recuperación inversa del diodo.

$di/dt$ : es el pico negativo de la intensidad.

$I_{rr}$ : es el pico negativo de la intensidad.

#### 2.1.4 Tiempo de recuperación directa:

$t_{fr}$  (tiempo de recuperación directo): es el tiempo que transcurre entre el instante en que la tensión ánodo-cátodo se hace positiva y el instante en que dicha tensión se estabiliza en el valor  $V_F$ .

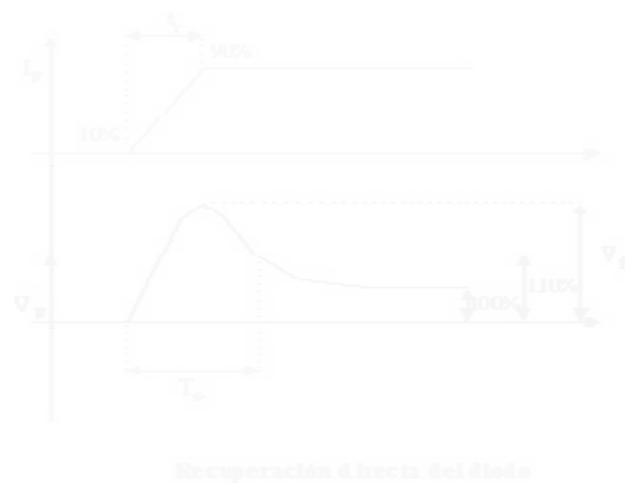


Figura 5. Recuperación directa del diodo

## 2.2. EL SCR

El tiristor (SCR) es un dispositivo semiconductor biestable de cuatro capas, PNPN figura 7, con tres terminales: ánodo (A), cátodo (K) y puerta (G), figura 6. Puede conmutar de bloqueo a conducción, o viceversa, en un solo cuadrante.

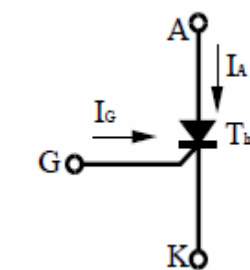


Figura 6, Símbolo

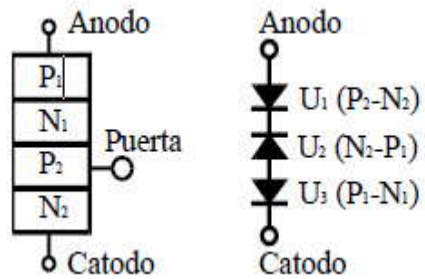


Figura 7.Union

La curva característica del SCR es la representada en la figura 8 donde:

- VDRM = Valor máximo de voltaje repetitivo directo.
- VRRM = Valor máximo de voltaje repetitivo inverso.
- $V_T$  = Caída de tensión de trabajo.
- $I_T$  = Intensidad directa de trabajo.
- $I_H$  = Intensidad de mantenimiento en estado de conducción.
- IDRM = Intensidad directa en estado de bloqueo.
- IRRM = Intensidad inversa en estado de bloqueo.

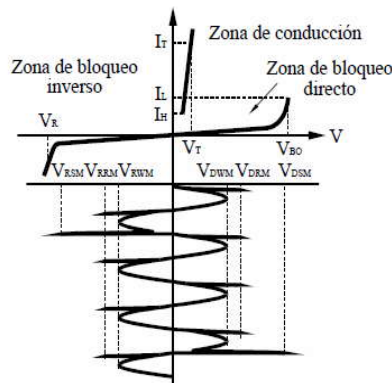


Figura 8.Curva característica

### 2.2.1 CARACTERÍSTICAS GENERALES

- Interruptor casi ideal.
- Soporta tensiones altas.

- Amplificador eficaz.
- Es capaz de controlar grandes potencias.
- Fácil controlabilidad.
- Relativa rapidez.
- Características en función de situaciones pasadas (memoria).

### 2.2.2 CARACTERÍSTICAS ESTÁTICAS

Corresponden a la región ánodo-cátodo. Son aquellos valores que determinan las posibilidades máximas de un determinado SCR. Estos datos son:

- Tensión inversa de pico de trabajo..... VRWM
- Tensión directa de pico repetitiva..... VDRM
- Tensión directa..... VT
- Corriente directa media..... ITAV
- Corriente directa eficaz..... ITRMS
- Corriente directa de fugas..... IDRM
- Corriente inversa de fugas..... IRRM
- Corriente de mantenimiento..... IH

Las características térmicas a tener en cuenta al trabajar con tiristores son:

- Temperatura de la unión .....Tj
- Temperatura de almacenamiento .....Tstg
- Resistencia térmica contenedor-disipador .....Rc-d
- Resistencia térmica unión-contenedor .....Rj-c
- Resistencia térmica unión-ambiente..... Rj-a
- Impedancia térmica unión-contenedor..... Rj-c

### 2.2.3 ÁREA DE DISPARO SEGURO

En este área figura 9. se obtienen las condiciones de disparo del SCR. Las tensiones y corrientes admisibles para el disparo se encuentran en el interior de la zona formada por las curvas:

- Curva A y B: límite superior e inferior de la tensión puerta-cátodo en función de la corriente positiva de puerta, para una corriente nula de ánodo.
- Curva C: tensión directa de pico admisible VGF.
- Curva D: hipérbola de la potencia media máxima PGAV que no debemos sobrepasar.

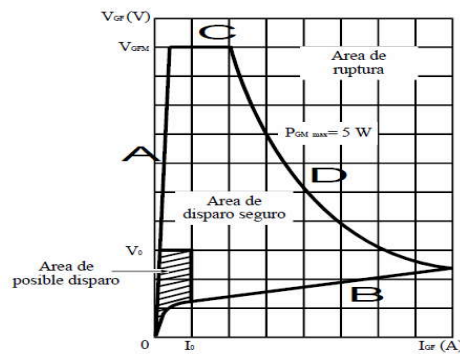


Figura 9. Característica de puerta del tiristor

## 2.2.4 Características Dinámica

- Tensiones transitorias:
  - Valores de la tensión superpuestos a la señal de la fuente de alimentación.
  - Son breves y de gran amplitud.
  - La tensión inversa de pico no repetitiva (VRSM) debe estar dentro de esos valores.
- Impulsos de corriente:
  - Para cada tiristor se publican curvas que dan la cantidad de ciclos durante los cuales puede tolerarse una corriente de pico dada.
  - A mayor valor del impulso de corriente, menor es la cantidad de ciclos.
  - La corriente y tensión media de un SCR dependen del ángulo de conducción como se puede observar en la figura 10.

- A mayor ángulo de conducción, se obtiene a la salida mayor potencia.
- Un mayor ángulo de bloqueo o disparo se corresponde con un menor ángulo de conducción:

$$\text{ángulo de conducción} = 180^\circ - \text{ángulo de disparo}$$

- Conociendo la variación de la potencia disipada en función de los diferentes ángulos de conducción podremos calcular las protecciones necesarias.

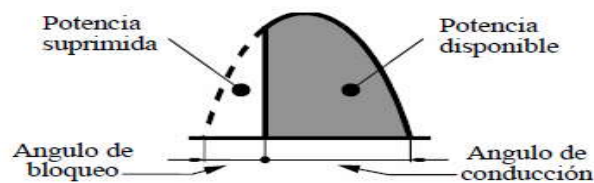


Figura 10. Ángulo de bloqueo y conducción de un tiristor

### 2.2.5 CARACTERÍSTICAS DE CONMUTACIÓN

Los tiristores no son interruptores perfectos, necesitan un tiempo para pasar de corte a conducción y viceversa. Vamos a analizar este hecho.

Tiempo de Encendido (TON): Tiempo que tarda el tiristor en pasar de corte a conducción.

- Tiempo de retardo ( $t_d$ ): tiempo que transcurre desde que la corriente de puerta alcanza el 50 % de su valor final hasta que la corriente de ánodo alcanza el 10 % de su valor máximo.
- Tiempo de subida ( $t_r$ ): tiempo necesario para que la corriente de ánodo pase del 10 % al 90 % de su valor máximo.

Tiempo de Apagado (TOFF): Tiempo que tarda el tiristor en pasar de conducción a corte.

- Tiempo de recuperación inversa ( $t_{rr}$ ): tiempo en el que las cargas acumuladas en la conducción del SCR, por polarización inversa de este, se eliminan parcialmente.
- Tiempo de recuperación de puerta ( $t_{gr}$ ): tiempo en el que, en un número suficiente bajo, las restantes cargas acumuladas se recombinan por difusión, permitiendo que la puerta recupere su capacidad de gobierno.

### 2.3 Rectificadores con Diodos o no Controlados

Los circuitos rectificadores no controlados, son convertidores de energía eléctrica de corriente alterna, a corriente continua, con valores de salida que dependen del valor de la tensión alterna de entrada y la carga conectada. En la figura 11, se muestra un circuito rectificador del tipo monofásico, con los bloques principales:

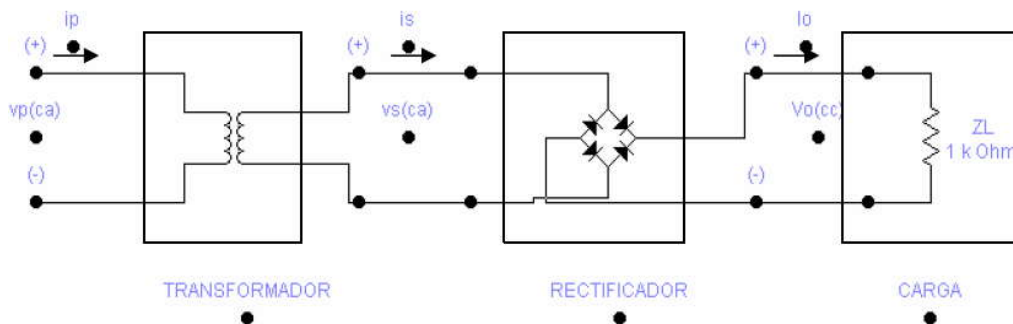


Figura 11. Circuito rectificador monofásico

Esta conversión, debe proporcionar una tensión continua de salida, con un mínimo de contenido armónico. Al mismo tiempo debe mantener la corriente de entrada tan sinusoidalmente como sea posible y en fase con la tensión de entrada, para que el factor de potencia se acerque a la unidad. En general, y para cualquier tipo de conversor de “energía eléctrica”, la calidad de conversión queda determinada por el “contenido armónico” de “la corriente de entrada, tensión de salida y corriente



de salida". Los contenidos armónicos de tensión y corriente, se pueden determinar mediante el desarrollo en serie de Fourier.

Tenemos varios tipos diferentes de circuitos rectificadores. Para su comparación respecto al mismo voltaje de suministro e igual "carga", resulta conveniente definir los "parámetros de rendimiento", para su evaluación y posterior selección.

### 2.3.1 Parámetros de rendimiento de la salida de los rectificadores:

- a) Tensión media o promedio de salida o tensión media en la carga:  $V_o$ ;  $V_{cd}$
- b) Corriente media o promedio de salida o corriente media en la carga:  $I_o$ ;  $I_{cd}$
- c) Potencia en continua sobre la carga:  $P_o = V_o \cdot I_o$ ;  $P_{cd} = V_{cd} \cdot I_{cd}$
- d) Valor medio cuadrático o tensión eficaz de salida:  $V_o(rms)$ .
- e) Valor medio cuadrático o corriente eficaz de salida:  $I_o(rms)$ .
- f) Potencia de salida  $P_o(rms) = V_o(rms) \cdot I_o(rms)$
- g) Eficiencia del rectificador:  $\eta(eta = P_o / P_o(rms) = P_{cd} / P_o(rms))$ .
- h) Tensión efectiva (rms) de la componente alterna de la tensión de salida:

$$V_{ca} = \sqrt{V_o(rms)^2 - V_o^2}$$

- i) Factor de forma de la tensión de salida:  $FF_v = V_o(rms) / V_o$
- j) Factor de forma de la corriente de salida:  $FF_i = I_o(rms) / I_o$
- k) Factores que determinan el contenido de componentes armónicos en la tensión de salida:
  - k1) Pulsación porcentual:  $(V_o(rms) / V_o) \cdot 100 = FF_v \cdot 100$  (factor de forma en %)
  - k2) Factor de pulsación:  $(V_{s1} / V_o) \cdot 100$ ;  $V_{s1}$  componente 1º armónica sobre la carga.
  - k3) Factor de componente ondulatoria o factor de rizado:  $RF(\gamma) \equiv (V_{ca} / V_o) \cdot 100$

$$RF \equiv \gamma \equiv (V_{ca} / V_o) \cdot 100 = (\sqrt{V_o(rms)^2 - V_o^2} / V_o) \cdot 100 = (\sqrt{(V_o(rms) / V_o)^2 - 1}) \cdot 100 =$$

$$RF \equiv \gamma \equiv (\sqrt{(FF_v)^2 - 1}) \cdot 100$$

L) Factor de utilización del transformador:

$$\text{TUF} \equiv P_o / V_s \cdot I_s$$

$V_s$ : tensión eficaz a la salida del transformador.

$I_s$ : Corriente eficaz a la salida del transformador.

El “factor de utilización” se define como la relación entre la potencia continua (promedio) sobre la carga ( $P_o$ ) y la potencia aparente del secundario del transformador. (al TUF, también se lo puede definir como  $P_o/V_p \cdot I_p$  o sea en función de la potencia aparente en la entrada del transformador). Este factor nos da una idea del empleo que se hace de la disponibilidad o aprovechamiento del arrollamiento del transformador. Para cada tipo de circuito rectificador tendremos un “TUF” distinto. Con este valor y la potencia continua sobre la carga, podremos determinar la potencia aparente del transformador y con este dato podemos determinar por cálculo, el volumen y costo del mismo. A igual potencia continua entregada a la carga, el volumen del transformador (ò potencia aparente) del transformador, dependerá del tipo de circuito rectificador seleccionado.

### 2.3.2 Parámetros de rendimiento respecto a la entrada de los rectificadores

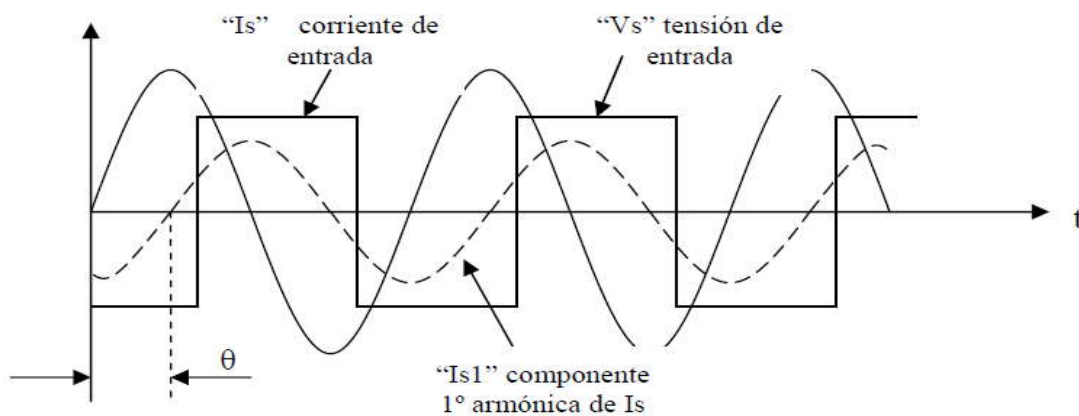


Figura 12. Parámetros de rendimiento

Estos parámetros quedan definidos en función de los contenidos armónicos y desfases entre la tensión de entrada ( $v_s$ ) y la corriente de entrada ( $i_s$ ), ver figura 12.

m) Factor de desplazamiento (DF): Si llamamos  $\theta$  al ángulo de desplazamiento o Desfasaje entre la tensión de entrada y la componente de 1º armónica o fundamental, de la corriente de entrada, entonces “DF” se define como:

$DF \equiv \cos \theta$  (a DF también se lo define como factor de potencia de desplazamiento  $DF \equiv DPF$ )

n) Factor armónico (HF): El factor armónico de la corriente de entrada se define como:

$$HF = \sqrt{(I_s^2 - I_{s1}^2) / I_{s1}^2} = \sqrt{(I_s^2 / I_{s1}^2) - 1}$$

Donde “ $I_s$ ” corresponde al valor eficaz (rms) de la corriente de entrada y “ $I_{s1}$ ” es el valor eficaz de la componente fundamental (1º armónica) de la corriente de entrada. El valor de “HF” es una medida de la distorsión de una forma de onda y también se lo conoce como THD (distorsión armónica total).

ñ) Factor de potencia de entrada (PF). Se lo define como:

$$PF \equiv [(V_s \cdot I_{s1}) / (V_s \cdot I_s)] \cdot \cos \theta$$

(Potencia activa 1º armónico/ potencia aparente total).

Si la corriente de entrada “ $I_s$ ” es sinodal, resulta  $I_s = I_{s1}$  y el factor de potencia “PF” es igual al factor de desplazamiento “DF”. Para una carga RL (resistencia e inductancia), el ángulo de desplazamiento “ $\theta$ ” se convierte en el “ángulo de impedancia”:

$$\theta = \tan^{-1}(\omega L / R).$$

o) Factor de cresta (CF) : Este factor nos da una medida de la corriente “pico máximo” de la entrada. Se define como:

$$CF \equiv I_{sp}(\text{pico}) / I_s(\text{rms})$$

Para un rectificador ideal debería cumplirse:

$$\eta = 100 \%$$

$$V_{ca} = 0 \text{ voltios}$$

$$FF_v = 1$$

$$R_F = \gamma = 0\%$$

$$HF = THD = 0$$

$$PF = DPF = 1$$

## 2.4 Rectificadores Controlados

Para obtener voltajes de salida controlados, se usan tiristores con control por fase en lugar de diodos. El voltaje de salida de los rectificadores de tiristor se varía controlando el ángulo de retardo o de disparo de los tiristores. Un tiristor controlado por fase se activa aplicando un pulso corto a su compuerta, y se desactiva por conmutación natural o de línea; en caso de que la carga sea muy inductiva, se desactiva disparando otro tiristor del rectificador durante el medio ciclo negativo del voltaje de entrada.

Estos rectificadores controlados por fase son sencillos y menos costosos, y su eficiencia es, en general, superior al 95%. Como convierten de CA a CD, a esos rectificadores controlados se les llama también convertidores CA-CD, y se usan en forma extensa en aplicaciones industriales, en especial en propulsores de velocidad variable, desde potencia fraccionaria hasta niveles de mega watts,

Los convertidores de control de fase se clasifican en dos tipos dependiendo de la fuente de alimentación:

Convertidores monofásicos:

Semiconvertidor

Convertidor completo

Convertidor dual

Convertidores trifásicos:

Semiconvertidor

Convertidor completo

Convertidor dual

Semiconvertidor: Es un convertidor de un cuadrante, y tiene una misma polaridad de voltaje y de corriente de salida.

Convertidor completo: Es un convertidor de dos cuadrantes, la polaridad de su voltaje de salida puede ser positiva o negativa. Sin embargo, la corriente de salida del convertidor completo sólo tiene una polaridad.

Convertidor dual: Puede operar en cuatro cuadrantes, y tanto su voltaje como su corriente de salida pueden ser positivos o negativos.

En algunas aplicaciones, los convertidores se conectan en serie para operar a mayores voltajes, y para mejorar el factor de potencia (PF) en la entrada. Para analizar los rendimientos de los convertidores controlados por fase con cargas RL, se puede aplicar el método de series de Fourier, en forma parecida a los diodos. Sin embargo, para simplificar el análisis se puede suponer que la inductancia de carga es suficientemente alta como para que la corriente en la carga sea continua y tenga rizo despreciable.

## **2.5 Principio de operación del convertidor controlado por fase**

Consideremos el circuito de la figura 13<sup>a</sup>. El cual tiene una carga resistiva. Durante el medio ciclo positivo del voltaje de alimentación, el ánodo del tiristor es positivo

con respecto a su cátodo, y se dice que el tiristor está *polarizado en forma directa*. Cuando se dispara el tiristor  $T$  en  $\omega t = \alpha$ , el tiristor  $T$  conduce y a través de la carga aparece el voltaje de entrada. Cuando el voltaje de entrada comienza a ser negativo en  $\omega t = \pi$ , el ánodo del tiristor es negativo con respecto a su cátodo, y se dice que el tiristor  $T$  tiene *polarización inversa* y se desactiva. El tiempo transcurrido después de que el voltaje de entrada comienza a ser positivo y se dispara el tiristor en  $\omega t = \alpha$ , se llama *ángulo de retardo*  $\alpha$ . De *disparo*  $\alpha$ .

La figura 13b. Muestra la región de operación del convertidor, donde el voltaje y la corriente de salida tienen una sola polaridad.

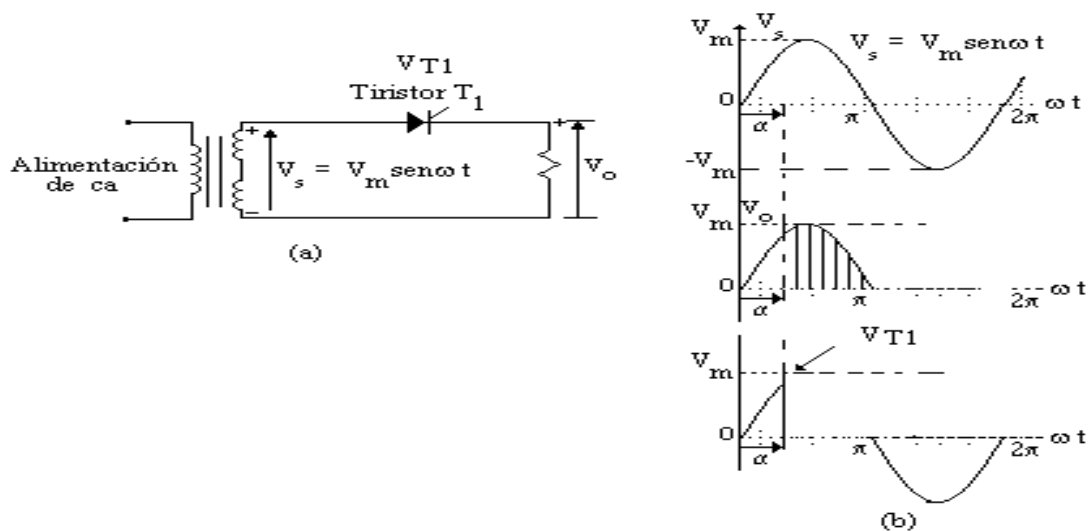


Figura 13.Región de operación del convertidor

Si  $V_m$  es el voltaje pico de entrada, el voltaje promedio de salida,  $V_{cd}$ , se determina con:

$$V_{cd} = \frac{1}{2\pi} \int_{\alpha}^{\pi} V_m \sin \omega t \, d\omega t = \frac{V_m}{2\pi} [ -\cos \omega t ]_{\alpha}^{\pi} = \frac{V_m}{2\pi} (1 + \cos \alpha)$$

V<sub>cd</sub> puede variar desde V<sub>m</sub>/π hasta 0, variando α de 0 a π. La tensión promedio de salida se vuelve máximo cuando α = 0, y el voltaje máximo de salida V<sub>dm</sub> es:

$$V_{dm} = V_m / \pi$$

Se normaliza el voltaje de salida con respecto a V<sub>dm</sub>, y el voltaje normalizado es

$$V_n = V_{cd} / V_m = 0.5(1 + \cos \alpha)$$

El voltaje raíz cuadrática media (rms) de salida es

$$\begin{aligned} V_{rms} &= \left[ \frac{1}{2\pi} \int_{\alpha}^{\pi} V_m^2 \sin^2 \omega t \, d(\omega t) \right]^{1/2} = \left[ \frac{V_m^2}{4\pi} \int_{\alpha}^{\pi} (1 - \cos 2\omega t) \, d(\omega t) \right]^{1/2} \\ &= \frac{V_m}{2} \left[ \frac{1}{\pi} \left( \pi - \alpha + \frac{\sin 2\alpha}{2} \right) \right]^{1/2} \end{aligned}$$

Secuencia de disparo. La secuencia de disparo para el tiristor es la siguiente:

1. Generar un pulso de señal en el cruce con cero positivo del voltaje de alimentación V<sub>S</sub>'
2. Retardar el pulso el ángulo α deseado y aplicarlo entre las terminales de compuerta y cátodo de T a través de un circuito de aislamiento de compuerta.

## 2.6 Semiconvertidor monofásico

La disposición del circuito de un Semiconvertidor monofásico aparece en la figura 14. Colocamos una carga altamente inductiva; la corriente de carga se supone continua y libre de componentes ondulatorias.

### 2.6.1 Operación:

Durante el medio ciclo positivo, el tiristor T<sub>1</sub> tiene polarización directa. Cuando el tiristor T<sub>1</sub> se dispara en ωt=α, la carga se conecta a la alimentación de entrada a través de T<sub>1</sub> y D<sub>2</sub> durante el período α ≤ ωt ≤ π.

Durante el período  $\pi \leq \omega t \leq (\pi + \alpha)$ , el voltaje de entrada es negativo y el diodo de marcha libre  $D_m$  tiene una polarización directa.  $D_m$  conduce para proporcionar la continuidad de corriente de la carga inductiva. La corriente de carga se transfiere de  $T_1$  y  $D_2$  a  $D_m$ , y el tiristor  $T_1$  así como el diodo  $D_2$  se desactivan.

Durante el ciclo negativo del voltaje de entrada el tiristor  $T_2$  queda con polarización directa, entrando en conducción en el instante  $\omega t = \alpha + a$ , hasta  $\omega t = 2\pi$  invirtiéndose entonces la polarización del diodo  $D_m$  y desactivándolo. La carga se conecta ahora a la alimentación a través de  $T_2$  y  $D_1$ .

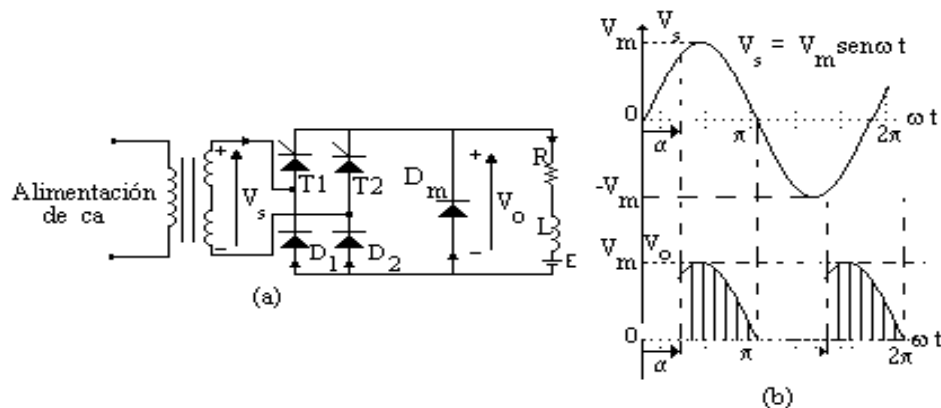


Figura 14. Semiconvertidor monofásico

La tensión promedio de salida se puede calcular a partir de

$$V_{do\alpha} = \frac{2}{2\pi} \int_{\alpha}^{\pi} V_M \sin \omega t \, d(\omega t) = \frac{2V_M}{2\pi} [-\cos \omega t]_{\alpha}^{\pi} = \frac{V_M}{\pi} [1 + \cos \alpha]$$

Pudiéndose variar  $V_{do}$  desde  $2V_m / \pi$  hasta 0, simplemente con variar  $\alpha$  entre 0 y  $\pi$ , la tensión medida máxima de salida  $V_{do}$  (sería para  $\alpha=0$ ) y el voltaje de salida medio normalizado  $V_n$  tendría por ecuaciones

$$V_{do} = 2V_m / \pi$$

$$V_n = \frac{V_{do\alpha}}{V_{do}} = 0.5 \times (1 + \cos \alpha)$$



La tensión eficaz de salida (o tensión rms) se determina a partir de:

$$V_{rms} = \sqrt{\frac{2}{2\pi} \int_{\alpha}^{\pi} V_m^2 \sin^2 \omega t \, d(\omega t)} = \sqrt{\frac{V_m^2}{2\pi} \int_{\alpha}^{\pi} (1 - \cos 2\omega t) \, d(\omega t)}$$

$$V_{rms} = \frac{V_M}{\sqrt{2}} \times \sqrt{\left[ \frac{1}{\pi} \left( \pi - \alpha + \frac{\sin 2\alpha}{2} \right) \right]}$$

Este tipo de convertidor presentara un mejor factor de potencia debido a la operación del diodo en marcha libre y es de uso común en aplicaciones de hasta 15kw, donde la operación en un solo cuadrante es todavía aceptable.

## 2.7 Convertidor Monofásico Completo

De la misma forma que hemos hecho con el rectificador de media onda, sustituyendo los diodos por tiristores para convertirlo en controlado. La figura 15 muestra tal circuito con carga resistiva. El ángulo de disparo,  $\alpha$ , es ahora el intervalo angular entre la polarización directa del tiristor y la aplicación de la señal de puerta. Si el ángulo de disparo es cero el circuito se comporta como un rectificador no controlado.

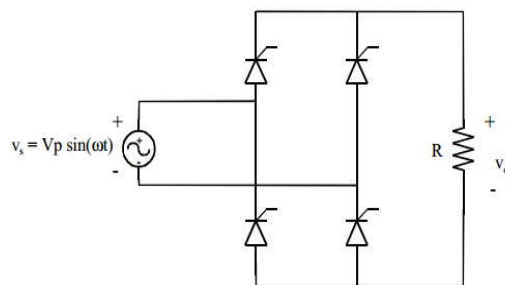


Figura 15. Convertidor monofásico

Funcionamiento con carga resistiva.

La figura 16 muestra las formas de onda de este rectificador. La tensión media en la carga es ahora el doble que en el circuito de media onda:

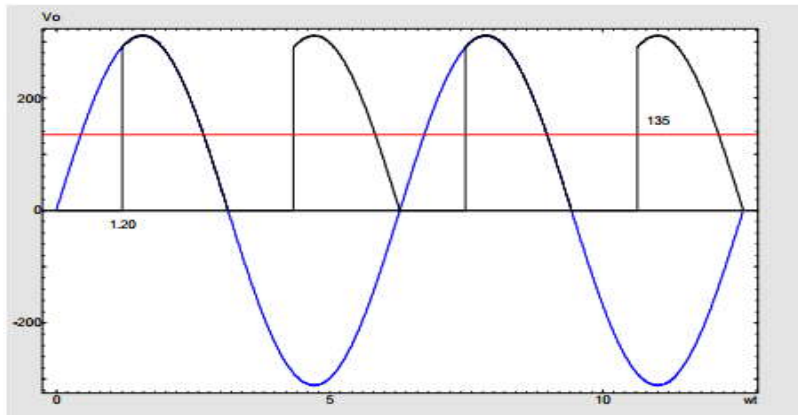


Figura 16. Forma de onda del rectificador monofásico

Donde

$$\langle V_o \rangle = \frac{1}{\pi} \int_{\alpha}^{\pi} V_p \sin(\omega t) d(\omega t) = \frac{V_p}{\pi} (1 + \cos(\alpha))$$

Y la potencia entregada a la carga  $I_{oRMSR}$ , vendrá dada a partir del cálculo de la corriente RMS por la carga:

$$I_{oRMS} = \sqrt{\frac{1}{\pi} \int_{\alpha}^{\pi} \left[ \frac{V_p}{R} \sin(\omega t) \right]^2 d(\omega t)} = \frac{V_p}{R} \sqrt{\left( \frac{1}{2} - \frac{\alpha}{2\pi} + \frac{\sin(2\alpha)}{4\pi} \right)}$$

Funcionamiento con carga RL.

En la figura 17a, se muestra el arreglo del circuito de un convertidor monofásico completo, con una carga muy inductiva, de tal modo que la corriente en la carga es continua y no contiene rizo. Durante el medio ciclo positivo, los tiristores T1 y T2 tienen polarización directa, y cuando se disparan en forma simultánea esos dos tiristores, en  $\omega t = \alpha$ , la carga se conecta a la fuente de alimentación a través de T1 y T2. Debido a la carga inductiva, los tiristores T1 y T2 continúan conduciendo después de  $\omega t = \pi$ , aun cuando el voltaje de entrada sea ya negativo. Durante el medio ciclo negativo del voltaje de entrada, los tiristores T3 y T4 tienen polarización directa, y el disparo de esos tiristores aplica el voltaje de alimentación a través de los tiristores T1 y T2 en forma de voltaje de bloqueo inverso. T1 y T2 se

desactivan por conmutación de línea o natural, y la corriente de carga es transferida de T1 y T2 a T3 y T4. La figura 17b, muestra las regiones de operación del

Convertidor, y la figura 17c, muestra las formas de onda del voltaje de entrada, voltaje de salida y corrientes de entrada y de salida.

La corriente de la carga puede ser nula durante algún instante o por el contrario no anularse en ningún momento, a estas dos situaciones se les conoce como modo de funcionamiento discontinuo y modo de funcionamiento continuo.

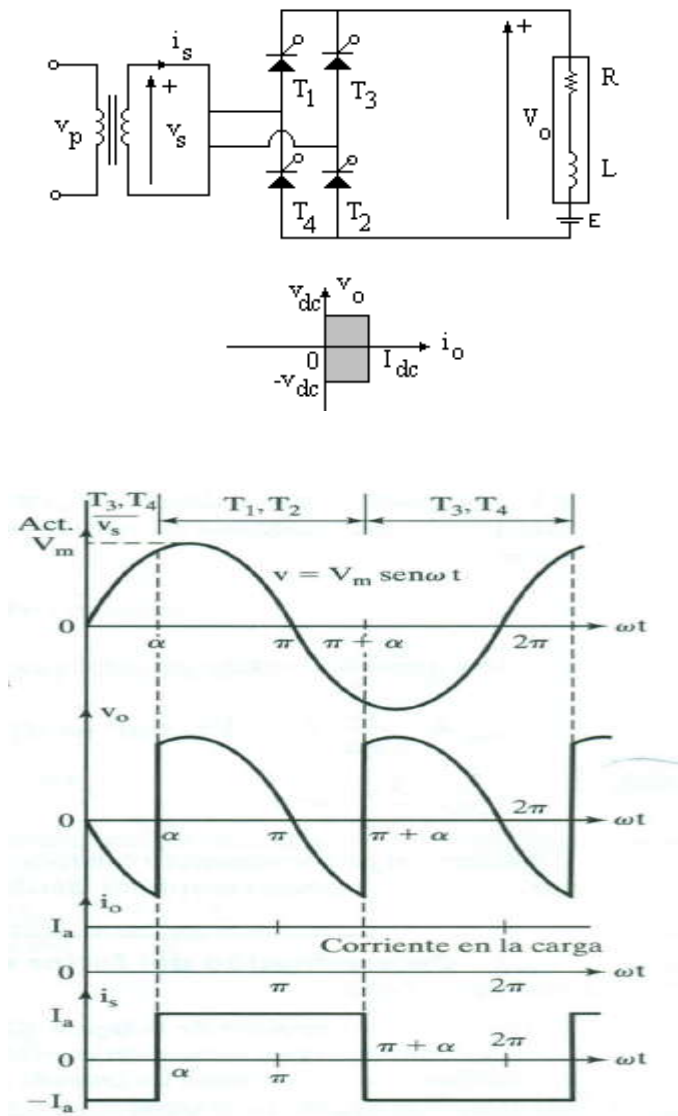


Figura 17. Entrada y salida del rectificador controlado

Durante el periodo de  $\alpha$  a  $\pi$ , el voltaje de entrada  $V_s$  y la corriente de entrada  $i_s$  son positivos, y la potencia pasa de la fuente a la carga. Se dice que el convertidor se opera en modo de rectificación. Durante el periodo de  $\pi$  a  $\pi + \alpha$ , el voltaje de entrada  $V_s$  es negativo y la corriente de alimentación  $i_s$  es positiva, y pasa potencia inversa de la carga a la fuente. Se dice que el convertidor está operado en modo de inversión. Este convertidor se usa en forma extensa en aplicaciones industriales. Dependiendo del valor de  $\alpha$ , el voltaje promedio de salida podría ser positivo o negativo, y proporciona una operación en dos cuadrantes. El voltaje promedio de salida se puede calcular a partir de:

$$\langle V_o \rangle = \frac{1}{\pi} \int_{\alpha}^{\pi+\alpha} V_p \sin(\omega t) d(\omega t) = \frac{2V_p}{\pi} \cos(\alpha)$$

Y  $V_{cd}$  se puede variar desde  $2V_m/\pi$  hasta  $-2V_m/\pi$ , haciendo variar a  $\alpha$  desde 0 hasta  $\pi$ . El voltaje promedio máximo de salida es  $V_{dm} = 2V_m/\pi$ , y el voltaje promedio normalizado de salida es

$$V_n = V_{cd} / V_{dm} = \cos \alpha$$

El valor RMS de la tensión de salida es:

$$V_{rms} = \left[ \frac{2}{2\pi} \int_{\alpha}^{\pi+\alpha} V_m^2 \sin^2 \omega t d(\omega t) \right]^{1/2} = \left[ \frac{V_m^2}{2\pi} \int_{\alpha}^{\pi+\alpha} (1 - \cos 2\omega t) d(\omega t) \right]^{1/2}$$

$$= \frac{V_m}{\sqrt{2}} = V_s$$

### 2.7.1 Modo de funcionamiento discontinuo:

En la figura 18, se muestra la tensión y la corriente de la carga en este modo de funcionamiento, el funcionamiento del circuito es idéntico al rectificador controlado de media onda con carga RL.

$$i(\omega t) = \frac{V_p}{Z} \left[ \sin(\omega t - \theta) - \sin(\alpha - \theta) e^{(\alpha - \omega t)/\omega\tau} \right] \quad \text{si} \quad \alpha \leq \omega t \leq \beta$$

$$Z = \sqrt{R^2 + (L\omega)^2} \quad \theta = \arctan\left(\frac{L\omega}{R}\right) \quad \omega\tau = \frac{L}{R}$$

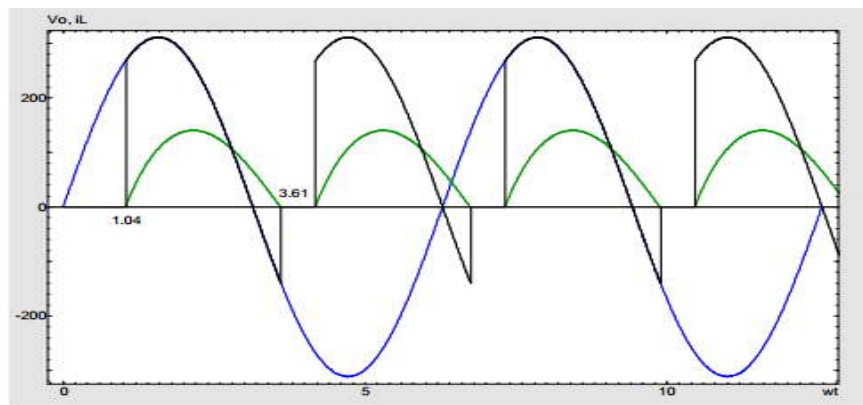


Figura 18. Tension y Corriente en la carga.

### 2.7.2 Modo de funcionamiento continuo:

Si se da el caso  $i(\pi + \alpha) \geq 0$  las ecuaciones anteriores ya no son válidas para describir la corriente por la carga, llegamos entonces a un estado estacionario en que la corriente de la carga será igual a la relación entre el valor medio de la tensión de salida y la resistencia de la carga. La condición de funcionamiento continuo vendrá dada por:

$$\sin(\pi + \alpha - \theta) - \sin(\alpha - \theta) e^{(\alpha - \pi - \alpha)/\omega\tau} \geq 0$$

Lo cual resulta en

$$\sin(\theta - \alpha)(1 - e^{-\pi/\omega\tau}) \geq 0$$

Para que se verifique la igualdad debe ser  $\alpha \leq \theta$  por lo que

$$\alpha \leq \arctg(L\omega / R)$$

Relación que puede usarse para saber si la carga es continua o discontinua.

El conjunto de armónicos por la serie de Fourier y la resolución del circuito en AC para cada uno de ellos. La aplicación de este concepto para el cálculo de la corriente RMS en la carga nos lleva a:

$$I_{RMS} = \sqrt{I_o^2 + \sum_{n=2,4,6,\dots}^{\infty} \left( \frac{I_n}{\sqrt{2}} \right)^2}$$

Y las amplitudes de los armónicos de corrientes viene dadas por

$$I_n = \frac{V_n}{|R + jL\omega|}$$

Donde  $V_n$  son los armónicos de la tensión de salida

$$V_n = \sqrt{a_n^2 + b_n^2}$$

$$a_n = \frac{2V_p}{\pi} \left[ \frac{\cos((n+1)\alpha)}{n+1} - \frac{\cos((n-1)\alpha)}{n-1} \right] \quad b_n = \frac{2V_p}{\pi} \left[ \frac{\sin((n+1)\alpha)}{n+1} - \frac{\sin((n-1)\alpha)}{n-1} \right]$$

$$n = 2, 4, 6, \dots$$

## 2.8 Lenguaje de Programación Java

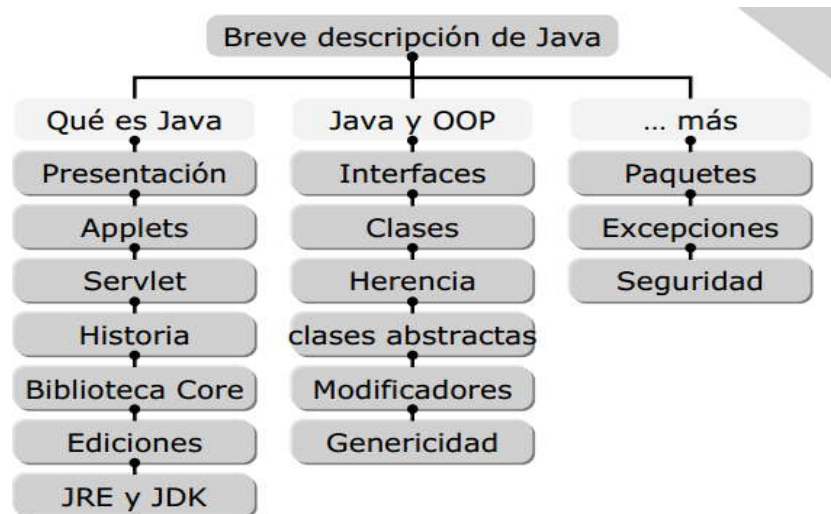


Figura 19. Arquitectura Java

JAVA es un lenguaje que deriva mucho de su sintaxis de C y C++, pero tiene menos facilidades de bajo nivel que cualquiera de ellos. Las aplicaciones de Java son generalmente compiladas a bytecode (clase Java) que puede correr en cualquier máquina virtual Java(JVM) sin importar la arquitectura de la computadora. Java es un lenguaje de programación de propósito general, concurrente, basado en clases, y orientado a objetos, que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Su intención es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo.

El lenguaje Java se creó con cinco objetivos principales:

1. Debería usar el paradigma de la programación orientada a objetos.
2. Debería permitir la ejecución de un mismo programa en múltiples sistemas operativos.

3. Debería incluir por defecto soporte para trabajo en red.
4. Debería diseñarse para ejecutar código en sistemas remotos de forma segura.
5. Debería ser fácil de usar y tomar lo mejor de otros lenguajes orientados a objetos, como C++.

Para conseguir la ejecución de código remoto y el soporte de red, los programadores de Java a veces recurren a extensiones como CORBA (Common Object Request Broker architecture), Internet communications Engine o OSGi respectivamente.

### **2.8.1 Orientado a objetos**

La primera característica, orientado a objetos (“OO”), se refiere a un método de programación y al diseño del lenguaje. Aunque hay muchas interpretaciones para OO, una primera idea es diseñar el software de forma que los distintos tipos de datos que usen estén unidos a sus operaciones. Así, los datos y el código (funciones o métodos) se combinan en entidades llamadas objetos. Un objeto puede verse como un paquete que contiene el “comportamiento” (el código) y el “estado” (datos). El principio es separar aquello que cambia de las cosas que permanecen inalterables. Frecuentemente, cambiar una estructura de datos implica un cambio en el código que opera sobre los mismos, o viceversa. Esta separación en objetos coherentes e independientes ofrece una base más estable para el diseño de un sistema software. El objetivo es hacer que grandes proyectos sean fáciles de gestionar y manejar, mejorando como consecuencia su calidad. Otra de las grandes promesas de la programación orientada a objetos es la creación de entidades más genéricas (objetos) que permitan la reutilización del software entre proyectos, una de las premisas fundamentales de la Ingeniería del Software.

La reutilización del software ha experimentado resultados dispares, encontrando dos dificultades principales: el diseño de objetos realmente genéricos es



pobremente comprendido, y falta una metodología para la amplia comunicación de oportunidades de reutilización. Algunas comunidades de “código abierto” (open Source) quieren ayudar en este problema dando medios a los desarrolladores para diseminar la información sobre el uso y versatilidad de objetos reutilizables y bibliotecas de objetos.

### **2.8.2 Independencia de la plataforma**

La segunda característica, la independencia de la plataforma, significa que programas escritos en el lenguaje Java pueden ejecutarse igualmente en cualquier tipo de hardware. Este es el significado de ser capaz de escribir un programa una vez y que pueda ejecutarse en cualquier dispositivo, tal como reza el axioma de Java, “write once, run anywhere”.

Para ello, se compila el código fuente escrito en lenguaje Java, para generar un código conocido como “bytecode” (específicamente Java bytecode) instrucciones máquina simplificadas específicas de la plataforma Java. Esta pieza está “a medio camino” entre el código fuente y el código máquina que entiende el dispositivo destino. El bytecode es ejecutado entonces en la máquina virtual (JVM), un programa escrito en código nativo de la plataforma destino (que es el que entiende su hardware), que interpreta y ejecuta el código. Además, se suministran bibliotecas adicionales para acceder a las características de cada dispositivo (como los gráficos, ejecución mediante hebras o threads, la interfaz de red) de forma unificada. Se debe tener presente que, aunque hay una etapa explícita de compilación, el bytecode generado es interpretado o convertido a instrucciones máquina del código nativo por el compilador JIT (Just In Time).

Las técnicas de compilación dan lugar a técnicas conocidas como:

compilación JIT (Just In Time, o “compilación al vuelo”), convierte el bytecode a código nativo cuando se ejecuta la aplicación.

Compilación' 'recompilación dinámica''' en la que la VM es capaz de analizar el comportamiento del programa en ejecución y recompila y optimiza las partes críticas. La recompilación dinámica puede lograr mayor grado de optimización que la compilación tradicional (o estática), ya que puede basar su trabajo en el conocimiento que de primera mano tiene sobre el entorno de ejecución y el conjunto de clases cargadas en memoria.

Compilación JIT y la recompilación dinámica permiten a los programas Java aprovechar la velocidad de ejecución del código nativo sin por ello perder la ventaja de la portabilidad en ambos.

La portabilidad es técnicamente difícil de lograr, y el éxito de Java en ese campo ha sido dispar. Aunque es de hecho posible escribir programas para la plataforma Java que actúen de forma correcta en múltiples plataformas de distinta arquitectura.

El concepto de independencia de la plataforma de Java cuenta, sin embargo, con un gran éxito en las aplicaciones en el entorno del servidor, como los Servicios Web, los Servlets, los Java Beans, así como en sistemas empotrados basados en OSGi, usando entornos Java empotrados.

### **2.8.3 El recolector de basura**

En Java el problema fugas de memoria se evita en gran medida gracias a la recolección de basura (o automatic garbage collector). El programador determina cuándo se crean los objetos y el entorno en tiempo de ejecución de Java (Java runtime) es el responsable de gestionar el ciclo de vida de los objetos. El programa, u otros objetos pueden tener localizado un objeto mediante una referencia a éste. Cuando no quedan referencias a un objeto, el recolector de basura de Java borra el objeto, liberando así la memoria que ocupaba previniendo posibles fugas. En definitiva, el recolector de basura de Java permite una fácil creación y eliminación de objetos y mayor seguridad.

## 2.8.4 Sintaxis

La sintaxis de Java se deriva en gran medida de C++. Pero a diferencia de éste, que combina la sintaxis para programación genérica, estructurada y orientada a objetos, Java fue construido desde el principio para ser completamente orientado a objetos. Todo en Java es un objeto (salvo algunas excepciones), y todo en Java reside en alguna clase (recordemos que una clase es un molde a partir del cual pueden crearse varios objetos).

- Todo en Java está dentro de una clase, incluyendo programas autónomos.
- El código fuente se guarda en archivos con el mismo nombre que la clase que contienen y con extensión “.java”. Una clase (class) declarada pública (public) debe seguir este convenio.
- El compilador genera un archivo de clase (con extensión “.class”) por cada una de las clases definidas en el archivo fuente. Una clase anónima se trata como si su nombre fuera la concatenación del nombre de la clase que la encierra, el símbolo “\$”, y un número entero.
- Los programas que se ejecutan de forma independiente y autónoma, deben contener el método “main()”.
- La palabra reservada “void” indica que el método main no devuelve nada.
- El método main debe aceptar un array de objetos tipo String. Por acuerdo se referencia como “args”, aunque puede emplearse cualquier otro identificador. La palabra reservada “static” indica que el método es un método de clase, asociado a la clase en vez de una instancia de la misma. El método main debe ser estático o “de clase”.
- La palabra reservada public significa que un método puede ser llamado desde otras clases, o que la clase puede ser usada por clases fuera de la jerarquía de la propia clase. Otros tipos de acceso son “private” o “protected”.

- La utilidad de impresión (en pantalla por ejemplo) forma parte de la biblioteca estándar de Java: la clase `System` define un campo público estático llamado `out`. El objeto `out` es una instancia de `PrintStream`, que ofrece el método `println (String)` para volcar datos en la pantalla (la salida estándar).
- Las aplicaciones autónomas se ejecutan dando al entorno de ejecución de Java el nombre de la clase cuyo método `main` debe invocarse. Por ejemplo, una línea de comando (en Unix o Windows) de la forma `java -cp . Hola` ejecutará el programa del ejemplo (previamente compilado y generado `Hola.class`). El nombre de la clase cuyo método `main` se llama puede especificarse también en el fichero `MANIFEST` del archivo de empaquetamiento de Java (`.jar`).

### **2.8.5 Applets**

Las applet Java son programas incrustados en otras aplicaciones, normalmente una página Web que se muestra en un navegador. Java incluye las clases sin tener que anteponer la ruta completa cada vez que se quieran usar en el código fuente.

### **2.8.6 Servlets**

Los servlets son componentes de la parte del servidor de Java EE, encargados de generar respuestas a las peticiones recibidas de los clientes.

### **2.8.7 Aplicaciones con ventanas**

Swing es la biblioteca para la interfaz gráfica de usuario de la plataforma Java SE.

Entornos de funcionamiento

El diseño de Java, su robustez, el respaldo de la industria y su fácil portabilidad han hecho de Java uno de los lenguajes con un mayor crecimiento y amplitud de uso en distintos ámbitos de la industria de la informática tales como:

**En dispositivos móviles y sistemas empujados**

**En el navegador web**

**En sistemas de servidor**

## **En aplicaciones de escritorio**

### **Plataformas soportadas**

## **2.9 Lenguaje de programación Python**

Es un lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis muy limpia y que favorezca un código legible.

Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, usa tipado dinámico y es multiplataforma.

Es administrado por la Python Software Foundation. Posee una licencia de código abierto, denominada Python Software Foundation License, que es compatible con la Licencia pública general de GNU.

### **2.9.1 Características:**

Python es un lenguaje de programación multiparadigma. Esto significa que más que forzar a los programadores a adoptar un estilo particular de programación, permite varios estilos:

- Programación orientada a objetos
- Programación imperativa
- Programación funcional.
- 

Python usa tipado dinámico y conteo de referencias para la administración de memoria.

Una característica importante de Python es la resolución dinámica de nombres; es decir, lo que enlaza un método y un nombre de variable durante la ejecución del programa (también llamado enlace dinámico de métodos).

Otro objetivo del diseño del lenguaje es la facilidad de extensión. Se pueden escribir nuevos módulos fácilmente en C o C++. Python puede incluirse en aplicaciones que necesitan una interfaz programable.

Aunque la programación en Python podría considerarse en algunas situaciones hostil a la programación funcional tradicional del Lisp, existen bastantes analogías entre Python y los lenguajes minimalistas de la familia Lisp como puede ser Scheme.

El intérprete de Python estándar incluye un *modo interactivo* en el cual se escriben las instrucciones en una especie de intérprete de comandos: las expresiones pueden ser introducidas una a una, pudiendo verse el resultado de su evaluación inmediatamente, lo que da la posibilidad de probar porciones de código en el modo interactivo antes de integrarlo como parte de un programa. Esto resulta útil tanto para las personas que se están familiarizando con el lenguaje como para los programadores más avanzados.

Python fue diseñado para ser leído con facilidad. Una de sus características es el uso de palabras donde otros lenguajes utilizarían símbolos. Por ejemplo, los operadores lógicos `!`, `||` y `&&` en Python se escriben `not`, `or` y `and`, respectivamente.

El contenido de los bloques de código (bucles, funciones, clases, etc.) es delimitado mediante espacios o tabuladores, conocidos como indentación, antes de cada línea de órdenes pertenecientes al bloque. Python se diferencia así de otros lenguajes de programación que mantienen como costumbre declarar los bloques mediante un conjunto de caracteres, normalmente entre llaves `{}`. Se pueden utilizar tanto espacios como tabuladores para indentar el código, pero se recomienda no mezclarlos.

**Python también tiene elementos de lenguaje los cuales son:**

**Comentarios**

**Variables**

**Tipos de datos**

**Listas y Tuplas**

**Diccionarios**

**Conjuntos**

**Listas por comprensión**

**Funciones**

**Clases**

**Condicionales**

**Bucles**

**Módulos**

## **2.10 Arquitectura cliente servidor en JAVA**

El término cliente/servidor es originalmente aplicado a la arquitectura de software que describe el procesamiento entre dos o más programas, una aplicación y un servicio soportante. Con respecto a la definición de arquitectura cliente/servidor se encuentran las siguientes definiciones:

- Cualquier combinación de sistemas que pueden colaborar entre sí para dar a los usuarios toda la información que ellos necesiten sin que tengan que saber dónde está ubicada.
- Es una arquitectura de procesamiento cooperativa donde uno de los componentes pide servicios a otro.
- Es un procesamiento de datos de índole colaborativo entre dos o más computadoras conectadas a una red.

- IBM define al modelo Cliente/Servidor. “Es la tecnología que proporciona al usuario final el acceso transparente a las aplicaciones, datos, servicios de cómputo o cualquier otro recurso del grupo de trabajo y/o, a través de la organización, en múltiples plataformas. El modelo soporta un medio ambiente distribuido en el cual los requerimientos de servicio hechos por estaciones de trabajo inteligentes o "clientes", resultan en un trabajo realizado por otros computadores llamados servidores”.
- “Es un modelo para construir sistemas de información, que se sustenta en la idea de repartir el tratamiento de la información y los datos por todo el sistema informático, permitiendo mejorar el rendimiento del sistema global de información”

## 2.10.2 ELEMENTOS PRINCIPALES

Como está claro los elementos principales de la arquitectura cliente servidor son justamente el cliente y el servidor. Por ejemplo dentro de un ambiente multimedia, el elemento cliente seria el dispositivo que puede observar el vídeo, cuadros y texto, o reproduce el audio distribuido por el elemento servidor.

### EN RESUMEN

C/S es una relación entre procesos corriendo en máquinas separadas El servidor (S) es un proveedor de servicios y El cliente (C) es un consumidor de servicios.

Los clientes y los servidores pueden estar conectados a una red local o una red amplia, bajo este modelo cada usuario tiene la libertad de obtener la información que requiera en un momento dado proveniente de una o varias fuentes locales o distantes y de procesarla como según le convenga. Los distintos servidores también pueden intercambiar información dentro de esta arquitectura.



## QUE ES UNA ARQUITECTURA

Una arquitectura es un entramado de componentes funcionales que aprovechando diferentes estándares, convenciones, reglas y procesos, permite integrar una amplia gama de productos y servicios informáticos, de manera que pueden ser utilizados eficazmente dentro de la organización como se muestra en la figura 20.

## QUE ES UN CLIENTE

Es el que inicia un requerimiento de servicio. El requerimiento inicial puede convertirse en múltiples requerimientos de trabajo a través de redes LAN o WAN. La ubicación de los datos o de las aplicaciones es totalmente transparente para el cliente.

## QUE ES UN SERVIDOR

Es cualquier recurso de cómputo dedicado a responder a los requerimientos del cliente. Los servidores pueden estar conectados a los clientes a través de redes LAN o WAN, para proveer de múltiples servicios a los clientes y ciudadanos tales como impresión, acceso a bases de datos, fax, procesamiento de imágenes, etc, como se puede apreciar en la figura 20.

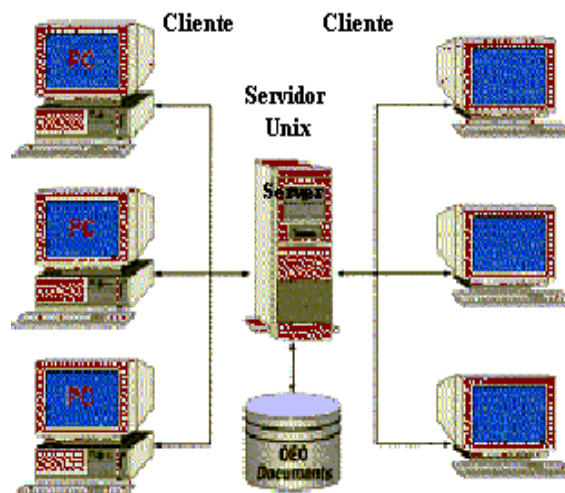


Figura 20. Arquitectura Cliente Servidor

## 2.10.2 ELEMENTOS DE LA ARQUITECTURA CLIENTE/SERVIDOR

En esta aproximación, y con el objetivo de definir y delimitar el modelo de referencia de una arquitectura Cliente/Servidor, debemos identificar los componentes que permitan articular dicha arquitectura, considerando que toda aplicación de un sistema de información está caracterizada por tres componentes básicos:

- Presentación /Captación de Información
- Procesos
- Almacenamiento de la Información

Los cuales se suelen distribuir tal como se presenta en la figura 21

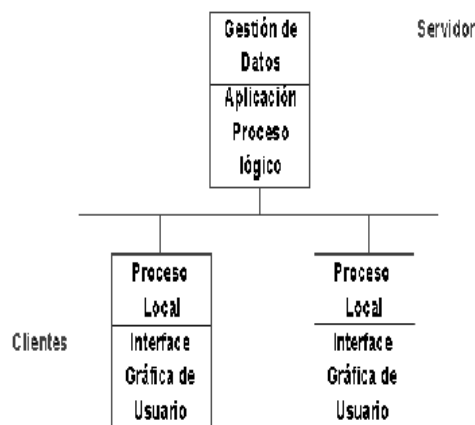


Figura 21. Distribución Cliente Servidor

### Aplicaciones Cliente/Servidor

Y se integran en una arquitectura Cliente/Servidor en base a los elementos que caracterizan dicha arquitectura, es decir:

- Puestos de Trabajo
- Comunicaciones
- Servidores

Tal como se presenta en la figura 22.

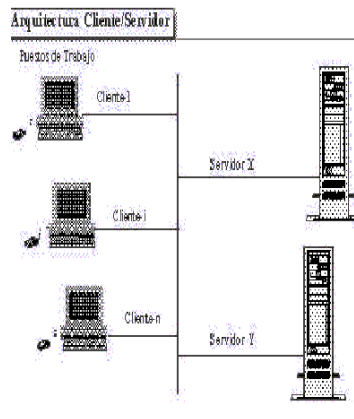


Figura 22. Aplicación arquitectura Cliente Servidor

De estos elementos debemos destacar:

#### El Puesto de Trabajo o Cliente

Una Estación de trabajo o microcomputador (PC: Computador Personal) conectado a una red, que le permite acceder y gestionar una serie de recursos» el cual se perfila como un puesto de trabajo universal. Nos referimos a un microcomputador conectado al sistema de información y en el que se realiza una parte mayoritaria de los procesos.

Debemos destacar que el puesto de trabajo basado en un microcomputador conectado a una red, favorece la flexibilidad y el dinamismo en las organizaciones. Entre otras razones, porque permite modificar la ubicación de los puestos de trabajo, dadas las ventajas de la red.

Las Comunicaciones en sus dos vertientes:

- Infraestructura de redes
- Infraestructura de comunicaciones

## Infraestructura de redes

Componentes Hardware y Software que garantizan la conexión física y la transferencia de datos entre los distintos equipos de la red.

## Infraestructura de comunicaciones

La arquitectura Cliente/Servidor es el resultado de la integración de dos culturas. Por un lado, la del Mainframe que aporta capacidad de almacenamiento, integridad y acceso a la información y, por el otro, la del computador que aporta facilidad de uso (cultura de PC), bajo costo, presentación atractiva (aspecto lúdico) y una amplia oferta en productos y aplicaciones.

## CARACTERISTICAS DEL MODELO CLIENTE/SERVIDOR

En el modelo CLIENTE/SERVIDOR mostrado en la figura 23, podemos encontrar las siguientes características:

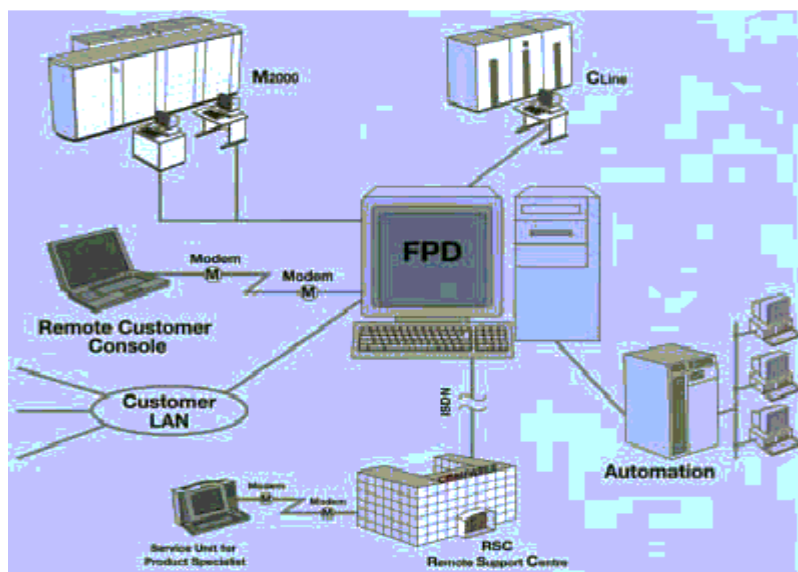


Figura 23. Modelo Cliente Servidor

1. El Cliente y el Servidor pueden actuar como una sola entidad y también pueden actuar como entidades separadas, realizando actividades o tareas independientes.
2. Las funciones de Cliente y Servidor pueden estar en plataformas separadas, o en la misma plataforma.
3. Un servidor da servicio a múltiples clientes en forma concurrente.
4. Cada plataforma puede ser escalable independientemente. Los cambios realizados en las plataformas de los Clientes o de los Servidores, ya sean por actualización o por reemplazo tecnológico, se realizan de una manera transparente para el usuario final.
5. La interrelación entre el hardware y el software están basados en una infraestructura poderosa, de tal forma que el acceso a los recursos de la red no muestra la complejidad de los diferentes tipos de formatos de datos y de los protocolos.
6. Un sistema de servidores realiza múltiples funciones al mismo tiempo que presenta una imagen de un solo sistema a las estaciones Clientes. Esto se logra combinando los recursos de cómputo que se encuentran físicamente separados en un solo sistema lógico, proporcionando de esta manera el servicio más efectivo para el usuario final.
7. Las funciones Cliente/Servidor pueden ser dinámicas.
8. Además se constituye como el nexo de unión más adecuado para reconciliar los sistemas de información basados en mainframes o minicomputadores

Su representación típica es un centro de trabajo (PC), en donde el usuario dispone de sus propias aplicaciones de oficina y sus propias bases de datos, sin dependencia directa del sistema central de información de la organización, al tiempo que puede acceder a los recursos de este host central y otros sistemas de la organización ponen a su servicio.

En conclusión, Cliente/Servidor puede incluir múltiples plataformas, por lo tanto, su implantación involucra diferentes tipos de estándares: APPC, TCP/IP, OSI, NFS, DRDA corriendo sobre DOS, OS/2, Windows o PC UNIX, en TokenRing, Ethernet, FDDI o medio coaxial, sólo por mencionar algunas de las posibilidades.

#### 2.10.4 TIPOS DE CLIENTES

En este punto se distinguen dos tipos de clientes ver figura 25 y figura 24, que son el cliente flaco y el cliente gordo.

“cliente flaco”: Servidor rápidamente saturado Gran circulación de datos de interface en la red.

“cliente gordo”: Casi todo el trabajo en el cliente, Gran circulación de datos inútiles en la red.

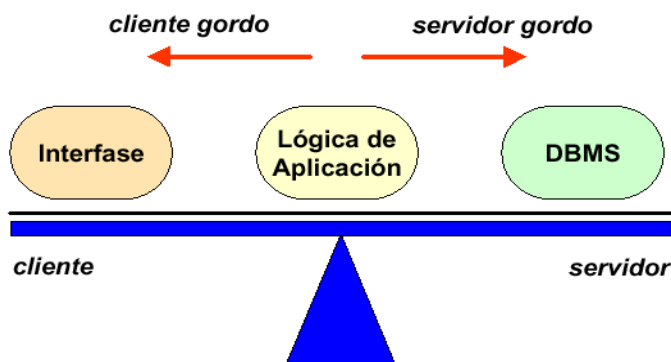


Figura 25. Cliente Gordo



Figura 24.  
Cliente Flaco

#### 2.10.5 TIPOS DE SERVIDOR

- **Servidores de archivos:** Servidor donde se almacena archivos y aplicaciones de productividad como por ejemplo procesadores de texto, hojas de cálculo, etc.

- **Servidores de bases de datos:** Servidor donde se almacenan las bases de datos, tablas, índices. Es uno de los servidores que más carga tiene.
- **Servidores de transacciones:** Servidor que cumple o procesa todas las transacciones. Valida primero y recién genera un pedido al servidor de bases de datos.
- **Servidores de Groupware:** Servidor utilizado para el seguimiento de operaciones dentro de la red.
- **Servidores de objetos:** Contienen objetos que deben estar fuera del servidor de base de datos. Estos objetos pueden ser videos, imágenes, objetos multimedia en general.
- **Servidores Web:** Se usan como una forma inteligente para comunicación entre empresas a través de Internet.

#### 2.10.6 ESTILOS DEL MODELO CLIENTE SERVIDOR

##### **Presentación Distribuida**

1. Se distribuye la interfaz entre el cliente y la plataforma servidora.
2. La aplicación y los datos están ambos en el servidor.
3. Similar a la arquitectura tradicional de un Host y Terminales.
4. El PC se aprovecha solo para mejorar la interfaz gráfica del usuario.

##### **Presentación Remota**

1. La interfaz para el usuario está completamente en el cliente.
2. La aplicación y los datos están en el servidor.

##### **Lógica Distribuida**

1. La interfaz está en el cliente.
2. La base de datos está en el servidor.
3. La lógica de la aplicación está distribuida entre el cliente y el servidor.

## **Administración De Datos Remota**

1. En el cliente residen tanto la interfaz como los procesos de la aplicación.
2. Las bases de datos están en el servidor.
3. Es lo que comúnmente imaginamos como aplicación cliente servidor

## **Base De Datos Distribuida**

1. La interfaz, los procesos de la aplicación, y parte de los datos de la base de datos están en cliente.
2. El resto de los datos están en el servidor.

## **Funciones de un Programa Servidor**

- Espera las solicitudes de los clientes.
- Ejecuta muchas solicitudes al mismo tiempo.
- Atiende primero a los clientes VIP.
- Emprende y opera actividades de tareas en segundo plano.
- Se mantiene activa en forma permanente.

### **2.11 Sistema de Procesamiento (Microcontrolador)**

Un microcontrolador es un circuito integrado, en cuyo interior posee toda la arquitectura de un computador, esto es CPU, memorias RAM, EEPROM, y circuitos de entrada y salida. Un microcontrolador debe ser programado para que se realice desde un simple parpadeo de un led hasta un sofisticado control. Un microcontrolador es capaz de realizar la tarea de muchos circuitos lógicos con compuertas AND, OR, NOT, NAND, Conversores A/D, D/A, temporizadores, decodificadores, etc, simplemente todo el diseño a una placa de reducido tamaño y pocos elementos. Este sistema consta de dos microcontroladores MC9S08QG8 de la familia Motorola freescale que permiten velocidades de ejecución de instrucciones alrededor de 300nS razón por la cual se escogió este



microcontrolador, figura 26 y la figura 27, muestra la arquitectura y distribución de los diferentes módulos especiales con los que cuenta este microcontrolador, como se mencionó el sistema consta de dos microcontroladores para la toma de señales, decisiones y comunicación.

### 2.11.1 EL MICROCONTROLADOR MC9S08QG8

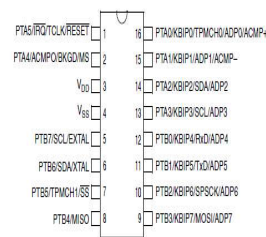


Figura 26. MC9S08QG8

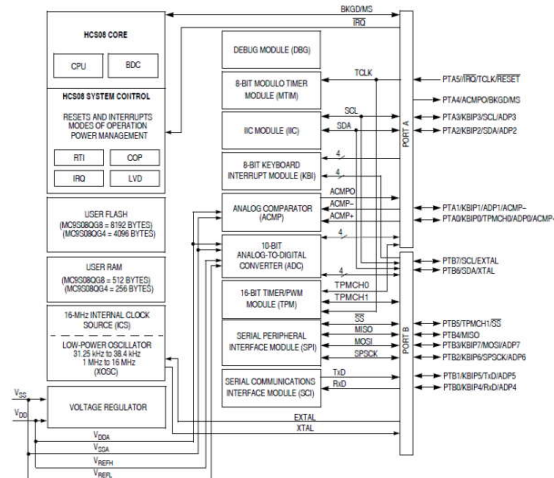


Figura 27. Arquitectura MC9S08QG8

Es uno de los microcontroladores más populares en la actualidad. Tiene las ventajas de incluir oscilador interno RC de 32MHz, MCLR programable, mayor capacidad de corriente, programación en bajo voltaje, etc.

## *Características Principales*

- \*Conjunto complejo de instrucciones (CISC).
- \*Oscilador interno de 32MHz
- \*Las instrucciones se ejecutan máximo en 5 ciclos de máquina excepto los saltos pero comúnmente usan solo 3 ciclos. Aquí hay que especificar que un ciclo de máquina se lleva 4 ciclos de reloj, si se utiliza el reloj interno de 32MHz, los ciclos de máquina se realizarán con una frecuencia de 4MHz, es decir que cada instrucción se ejecutará en 125nS
- \*Memoria de programa: 8192 locaciones de 14 bits
- \*Memoria de datos: Memoria RAM de 512 bytes (8 bits por registro)
- \*Memoria EEPROM: 128 bytes (8 bits por registro)
- \*Stack de 16 niveles
- \*16 Terminales de I/O que soportan corrientes de hasta 25 mA
- \* 2 Temporizadores
- \* Módulo de interrupción de lectura por teclado con 8 canales
- \* Módulos de comunicación serie, comparadores, PWM

Otra característica de los Motorola Freescale es el manejo de los bancos de registros, en línea general, los registros se clasifican como de uso general (GPR) y de uso

Específico o de funciones especiales (SFR). Los registros de uso general pueden ser usados directamente por el usuario, sin existir restricciones. Pueden servir para almacenar resultados aritmético-lógicos que se reciben desde el registro Acc (acumulador), datos que provienen de las puertas de entradas, etc.

Los registros de uso específicos no pueden ser usados directamente por el usuario. Estos registros controlan prácticamente todo el funcionamiento del microcontrolador, pues toda la configuración necesaria para funcionamiento del microcontrolador es hecho a través de algún tipo de SFR.

Los módulos usados en este trabajo son:

- Keyboard Interrupt (S08KBIV2)
- Timer/Pulse-Width Modulator (S08TPMV2)
- Interrupt Pin Request Status and Control Register (IRQSC)
- Analog-to-Digital Converter (S08ADC10V1)

### **Módulo de Interrupción Externa (IRQSC)**

El Pin IRQ (interrupción externa), que comparte con PTA5 (entrada de propósito general) e interrupción por teclado (KBI) en el S08QG8, provee una interrupción enmascarable. La Figura XXX muestra el diagrama en bloques completo del MCU, en donde se resalta el Módulo IRQ y el pin de entrada correspondiente.

### **Características**

Entre las características del IRQ se incluyen:

- Pin de interrupción externa, IRQ
- Bits de control de interrupción IRQ
- Detección de Interrupción programable por flanco solamente o por flanco y nivel.
- Detección de interrupción automática
- Pullup interno seleccionable.

### **Descripción Funcional**

La funcionalidad del pin IRQ se activa por ajuste del registro de configuración 2 (CONFIG2), específicamente el bit IRQEN. Un cero desactiva la función IRQ y el pin PTA2 tomará las demás funcionalidades. Un uno habilita la función IRQ.

Un nivel bajo aplicado al pin de requerimiento de interrupción externa (IRQ) puede generar un requerimiento de interrupción del CPU. La Figura 28 muestra la estructura del módulo IRQ. Las señales de Interrupción sobre el pin IRQ se

mantienen en el LATCH IRQ (cerrojo IRQ). El LATCH IRQ se mantiene hasta que ocurra una de las siguientes acciones:

- Vector de carga IRQ — Un vector de carga IRQ genera automáticamente una señal de acuse de interrupción que borra el LATCH IRQ.
- Borrado por Software, el Software puede borrar el LATCH IRQ escribiendo un 1 al bit ACK en el registro de control y estado de interrupción (INTSCR).
- Reset — Al resetear se borra automáticamente el LATCH IRQ.

El pin de interrupción externa se dispara en un flanco descendente de reset fuera de un reset y es configurable por software para ser detectado por flanco descendente o por flanco descendente y nivel bajo. El bit MODE en INTSCR controla la sensibilidad de disparo del pin IRQ.

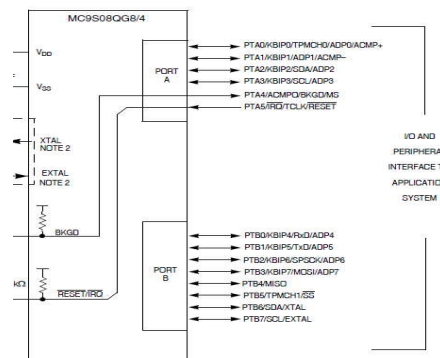


Figura 28. Módulo IRQ pin

Cuando se configura, el bit IMASK en las máscaras INTSCR se genera el requerimiento de interrupción de IRQ. No se almacenará un requerimiento de interrupción en la lógica de prioridad de interrupción a menos que IMASK sea borrado.

Se debe tener en cuenta que la máscara de interrupción como se observa en la figura 29, en el registro de código de condición (CCR) enmascara todos los requerimientos de interrupciones, incluyendo el requerimiento de interrupción IRQ.

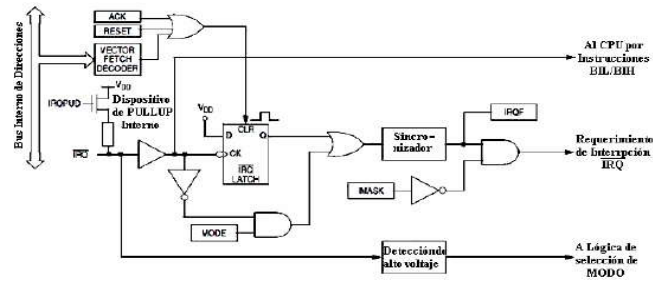


Figura 29. Mascara de Interrupción

## Módulo de Interrupción por Teclado (Keyboard Interrupt (S08KBIV2))

El módulo de interrupción por teclado permite disponer de ocho interrupciones externas independientes, accesibles a través de los pines 0 a 7 del puerto A (PTA0 a PTA3 y PTB0 a PTB3 ), ver figura 30.

### Características

Este módulo incluye:

- Ocho pines de interrupción por teclado con bits de habilitación separados y uno de enmascaramiento de interrupción.
- Dispositivo de pull-up configurable por software en el caso de que el bit del puerto sea configurado como entrada.
- Sensibilidad de la interrupción programable a flanco solo o flanco y nivel.
- Salida desde modos de bajo consumo.

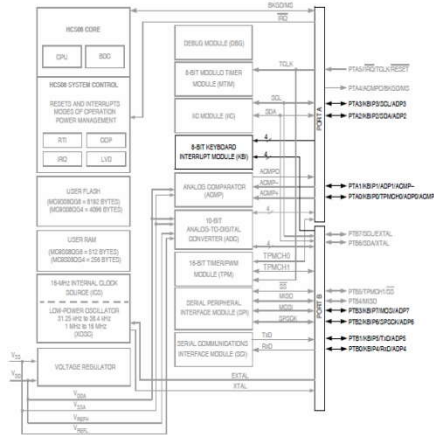


Figura 30. Puertos Keyboard

## Descripción funcional

El módulo de interrupción por teclado controla la habilitación y deshabilitación de las funciones de interrupción sobre seis pines del puerto A de forma independiente uno del otro.

## Operación de Teclado

Escribiendo los bits KBIE0-KBIE7 en el registro de habilitación de interrupción por teclado, se habilita o deshabilita, independientemente, cada pin del puerto A como un pin de interrupción por teclado, esto activa automáticamente el respectivo dispositivo interno de pull-up desde el bit del registro de habilitación de entradas del puerto A y B (PTxPUE<sub>x</sub>).

Una interrupción por teclado se produce cuando una o más de éstas entradas se pone en estado bajo —0 lógico después de estar todas en estado alto (1 lógico).

El bit MODEK del registro de control y estado de teclado, controla el modo de disparo de la interrupción por teclado:

- Si la interrupción por teclado es solo sensible a flanco, un flanco de bajada sobre el pin no almacena un pedido de interrupción si otro pin ya está en nivel bajo. Para prevenir la pérdida de un pedido de interrupción, el software puede deshabilitar la última entrada mientras está en nivel bajo.

- Si la interrupción por teclado es sensible al flanco descendente y nivel bajo, se producirá un requerimiento de interrupción cuando cualquier entrada de interrupción esté en nivel bajo, la figura 31 muestra la mascara de interrupción.

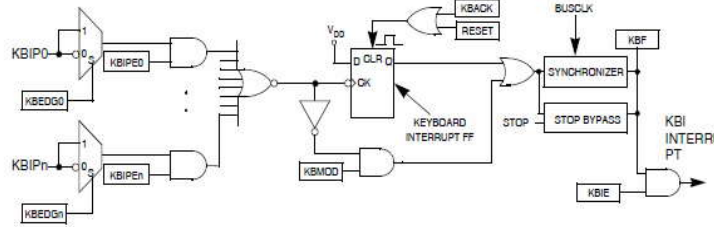


Figura 31. Mascara interrupción KBD

### Módulo Conversor Analógico/Digital (Analog Comparator (S08ACMPV2))

Antes de la descripción del módulo ADC del microcontrolador se analizarán las características y la utilidad de los conversores AD, Como se describió anteriormente, los dispositivos Microcontroladores son puramente digitales, lo que no quita que puedan procesar señales analógicas. Para el procesamiento de señales analógicas en los MCUs se utilizan los Conversores AD, lo cuales digitalizan las señales analógicas para su posterior procesamiento dentro del microprocesador.

El proceso de convertir introduce una inevitable pérdida de información. Esta pérdida es inherente al proceso de discretizar las señales análogas y continuas, que finalmente serán llevadas a cantidades binarias.

La Figura 32, representa una señal análoga continua entre los puntos  $t_0$  y  $t_1$ , que desde el punto de vista de la magnitud será sometida a un número discreto de valores binarios y que la cantidad de valores se conoce con el nombre de Resolución del sistema.

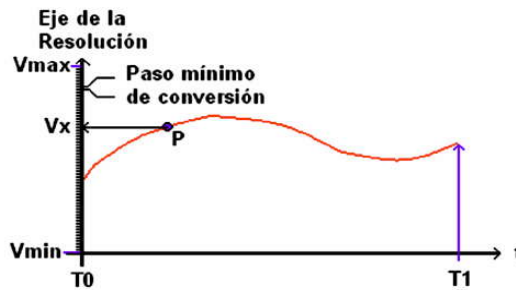


Figura 32. Señal Análoga

La Resolución de un conversor analógico al digital es la cantidad de valores, discretos, en los cuales se interpreta la señal a digitalizar. Por ejemplo, para un procesador con un conversor A/D que tiene una resolución de 12 bits el número de valores discretos en los cuales se puede valorar a una señal, sería de  $2^{12} = 4096$ . El valor ideal para esos valores sería un número infinito, pero tecnológicamente es imposible.

Esos valores deben estar comprendidos dentro de dos límites, que forman la ventana de conversión o valores de referencia ( $V_{min}$ ,  $V_{max}$ ), el punto P tiene una interpretación en el mundo de lo discreto y es de  $V_x$ , se recomienda que el usuario aproveche al máximo la resolución del sistema, adecuando la señal analoga para que excursión de la manera más completa en la ventana de conversión. Por ejemplo, una señal con un valor máximo de 100mV deberá ser amplificada por un factor de 30, para una ventana de conversión de 3V y de ésta manera aprovechar la resolución del sistema.

Para calcular el paso mínimo de conversión y por otro lado conocer el intervalo de pérdida de información, supóngase que se tiene una señal sometida a un conversor de 12 bits de resolución, un  $V_{min} = 0V$  y un  $V_{max} = 5V$ . El paso mínimo de conversión está dado por:

- Paso mínimo =  $(V_{max} - V_{min}) / \text{Resolución (pasos)}$
- Paso mínimo =  $(5V - 0V) / 2^{12}$
- Paso mínimo = 1.22mV



El cálculo anterior indica que la diferencia en magnitud entre el resultado de una conversión y la inmediatamente superior (o inferior) es de 1.22mV. Todo valor que no sea múltiplo entero de un paso mínimo, se deberá aproximar al valor más cercano y es allí donde un conversor A/D ignora información del mundo analógico.

El Muestreo (sample) es otra característica importante de un conversor A/D y se refiere a la cantidad de muestras en la unidad de tiempo que se pueden procesar y convertir a cantidades discretas.

La Figura 33, presenta una señal analógica continua entre los puntos  $t_0$  y  $t_1$ , que desde el punto de vista del muestreo es sometida a un número finito de muestras y que cada muestra es tomada a un intervalo constante  $T$ , llamado período de muestreo.

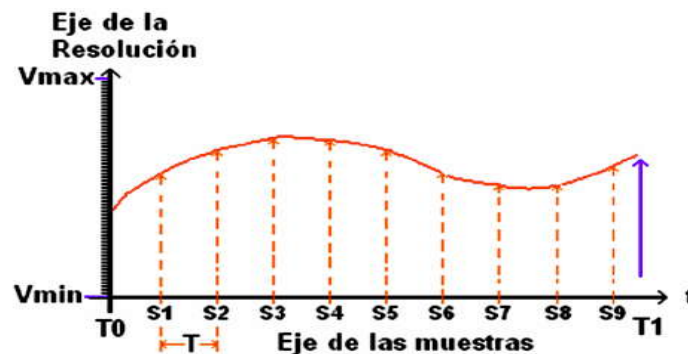


Figura 33. Muestreo del Conversor ADC

Al igual que en la resolución el muestreo introduce pérdida de información, debido a los valores que no son muestreados entre dos intervalos de muestreo contiguos. Idealmente la tasa de muestreo debería ser infinita, pero existen restricciones tecnológicas. Entonces, mientras menor sea la separación entre los  $s_i$  ( $T$  pequeño), mostrados en la Figura 33, más fiel será la señal digitalizada con respecto a la señal analógica original.

## Descripción

El ADC en el Microcontrolador M68HC08 es un conversor unipolar, de aproximaciones sucesivas disponible para una resolución de 8 bits o 8-/10-bits, garantizado para ambas opciones. Los Microcontroladores de Freescale S08QG8 incluyen módulos ADC con una gran variedad de opciones para el usuario, permitiendo de este modo diferentes aplicaciones.

### **Algunas de las características del módulo ADC incluyen:**

- De 4 a 24 canales ADC con una gran variedad de opciones de por medio.
- Dos opciones para la resolución: están disponibles los módulos ADC de 8 bits o 8-/10-bits, dependiendo de la familia del Microcontrolador.
- El tipo de conversión es adaptable a cada aplicación: Permite conversión de una muestra o continua, con la opción de conversión auto-scanen algunos Microcontroladores.
- Soporta ambos métodos de programación: Incluye una conversión completa por flag y una conversión completa por interrupciones, permitiendo al usuario elegir un método por llamada selectiva o uno basado en interrupciones.
- Frecuencia seleccionable del reloj ADC: Incluye un preescalador de reloj, y algunos Microcontroladores incluyen la opción de elegir entre el reloj de bus o el cristal externo como reloj de entrada.
- Flexibilidad: El ADC 8-/10-bits contiene cuatro opciones para justificar el resultado de la conversión (10 bits justificados a izquierda, 10 bits justificados a derecha, 10 bits justificados a izquierda con signo, y el modo de truncado de 8 bits).

## **Los principales registros del módulo ADC son:**

- Status and control register, ADCSC
  - Flags (banderas) para una conversión ADC completa
  - habilitación de interrupciones ADC
  - Selección de conversiones continuas ADC o una muestra ADC
  - Selección de uno de los canales ADC para ser escaneado
- Data result registers, ADCRH and ADCRL: Se actualiza con el resultado cada vez que un ADC termina la conversión, y puede ser:
  - a. De un byte de largo para ADC de 8 bits.
  - b. De dos bytes de largo para ADCs 8-/10-bits conteniendo los datos en el formato seleccionado (ADCRH-ADCRCL o ADCRH0:EI ADRL0)
- Compare value registers, ADCCVH and ADCCVL
- Configuration register, ADCCFG
- Pin enable registers, APCTL1, APCTL2, APCTL3

## **Descripción del Módulo ADC en el MCU S08QG8**

Existen 8 canales que pueden ser utilizados como entrada del conversor a través de los pines PTA0 hasta PTA7, un multiplexor analógico permite seleccionar uno de los ocho canales como entrada del conversor (ADCVIN). ADCVIN es convertido por el método de aproximaciones sucesivas. La resolución del conversor es de 10 bits, cuando se realiza la conversión, el resultado es escrito en el registro de datos ADCR y configura un flag o genera una petición de interrupción. Los pines de E/S de propósitos generales PTAxson compartidos con el conversor como canales de entrada de la señal a medir. Seleccionando uno de los canales como entrada del conversor a través del registro ADCR, la función del pin como pin de E/S deja de tener efecto. El resto de los pines son controlados por

la lógica del puerto de E/S y podrán ser utilizados como pines de E/S de propósito general.

### **Tensión de conversión**

La entrada del conversor debe estar entre Vdd y Vss para que la conversión sea lineal. Si la entrada supera Vdd, el resultado de la conversión siempre será \$3FF y si disminuye por debajo de Vss el resultado será \$000. La señal de entrada debe mantenerse estable durante el tiempo de conversión.

### **Tiempo de conversión**

El conversor requiere 16 ó 17 (si no existe sincronismo entre el reloj del Bus y el del conversor) ciclos de reloj del conversor para completar una conversión. La conversión comienza en el primer flanco ascendente inmediatamente después de seleccionar el canal, escribiendo en el registro ADCSC.

.

$$\text{Tiempo de conversión} = 16 \text{ ciclos de reloj ADC} / \text{Frecuencia de reloj ADC}$$

### **Módulo de Interfaz del Temporizador (Timer/Pulse- WidthModulator (S08TPMV2))**

Esta sección describe el Módulo de interfaz de Temporizador (TPM). El TPM es un temporizador de dos canales, que proporciona referencia de tiempo con **captura de entrada (input capture)**, **comparador de salida (output compare)**, y funciones de **modulación por ancho de pulso PWM**. La Figura 34, muestra un diagrama en bloques del TPM.



## Preescalador del contador del TPM

Figura 35. Diagrama de Bloques TPM

## Captura de Entrada

En este modo el MCU medirá eventos temporales externos, aplicados a uno de los pines de los canales del TPM. Estos eventos pueden ser tanto, la medida del ancho de un pulso o la frecuencia de una señal, la figura 36, muestra un esquema del funcionamiento de este modo. Una aplicación posible puede ser la lectura de pulsos de un encoder incremental para medir la velocidad de giro de un motor de corriente continua, otra aplicación puede ser la medida del ancho de pulso de un paquete de datos enviado por otro dispositivo, para determinar a qué frecuencia está transmitiendo y así poder recibir los datos a la misma frecuencia. Con la función de captura de entrada, el TPM puede capturar el momento en que ocurre un evento externo. Cuando ocurre un flanco activo sobre el pin de un canal de captura de entrada, el TPM almacena el contenido del contador en el registro del canal del TPM, TPMxH:TPMxL. La polaridad del flanco activo es programable. La captura de entrada del TPM puede generar requerimientos de interrupción de la unidad central de procesos (CPU).

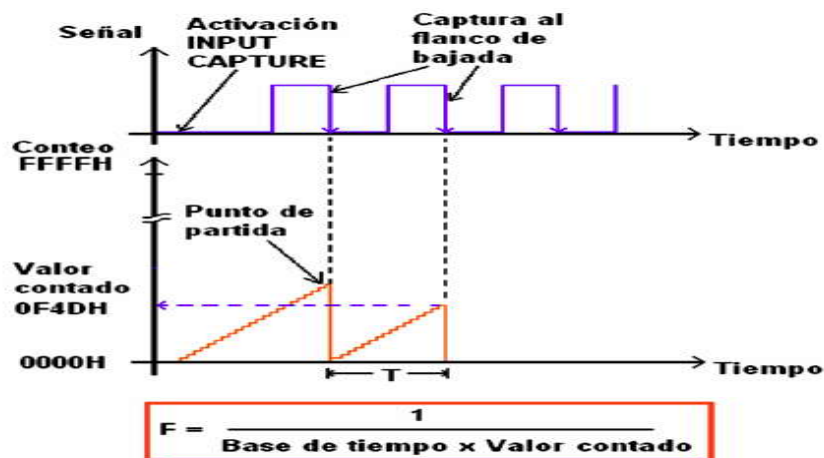


Figura 36. Función Input Capture

### Capítulo 3. Diseño y construcción de rectificador monofásico controlado y control de fase monofásico

Se construyeron dos módulos con conectores tipo RCA así:

- Modulo Rectificador Monofásico no Controlado, Semicontrolado y Controlado.
- Módulo de Control de fase monofásico

#### 3.1 MODULO CONTROL CA – CC MONOFASICO

Este módulo consta de elementos de hardware especiales como sensor de Tensión, sensor de Corriente, circuito integrado de aplicación específica (ASIC) para establecer comunicaciones de alta velocidad RS232, control de ángulo de disparo remoto, control de ángulo de disparo manual por medio de dos pulsadores, sistemas microcontrolados de procesamiento de la información. Como se muestra a continuación en la figura 37.

Este módulo también cuenta con 4 diodos para implementar topologías de puente rectificado no controlado, además se cuenta con una disposición de 4 SCR's para las diferentes topologías de rectificación controlada y semicontrolado.



Figura 37. A Modulo Control CA – CC Monofásico





Figura 38. B Modulo Control CA – CC Monofásico

### 3.1 .1 Control On-Line

En primer lugar se debe descargar los software de programación denominados NetBeans y Python que son totalmente gratuitos y sirven para sistemas operativos como Windows, Linux y Mac.

Para realizar un control en lazo cerrado on-line de una carga (Planta) se implementó en el software un espacio para código libre como lo es el mismo Java. En este lugar de trabajo solo se necesita conocer la referencia deseada y la estrategia a seguir que como se mencionó anterior mente puede ser un PI, pues los parámetros de tensión y corriente siempre están a disposición en dos variables denominadas TENSION Y CORRIENTE.

Se debe ingresar la librería del programa con la siguiente ruta: después de instalar los dos programas necesarios para que la aplicación funcione se debe realizar el siguiente recorrido

- Buscar el nombre del proyecto que es HERMES
- Desplegar el icono de Source Packages
- Abrir carpeta denominada hermes

- Abrir la clase HermesView.java
- Desplegar el hilo hiloEnviarDatosCalculados

Donde el resultado de la acción de control es el ángulo de disparo de los tiristores del sistema controlado o semicontrolado que determina la tensión promedio de alimentación a la carga

La conexión desde cualquier lugar del mundo se realiza por medio de la configuración del cliente el cual debe contar con los siguientes parámetros:

- La conexión VPN
- La dirección IP del servidor (Host) a la red

Para poder acceder al host por medio de escritorio Remoto desde afuera de la Universidad es necesario realizar una conexión VPN, los pasos para realizar esta conexión son los siguientes:

- Ingresar por medio de un navegador a la dirección <https://200.21.217.1:10443>
- Las credenciales son las siguientes:  
 Usuario: jovan  
 Contraseña: B0050SX89W
- Una vez haya ingresado, en la sección Tunnel Mode dar click en un enlace para descargar el software gratuito que permite generar la conexión entre cliente servidor, luego aparecerá un botón denominado Connect, cuando se haya conectado con éxito puede realizar la conexión del escritorio remoto a la maquina con IP 192.168.11.212.

Los parámetros anteriores fueron solicitados a la red de la universidad para una conexión cliente servidor se necesita n como elemento fundamental una dirección estática, ya que comercialmente la mayoría de la conexiones son de tipo dinámica,

o que significa que la maquina constantemente está cambiando de dirección IP y nunca encontraremos en la nube el host.

### 3.2 MODULO CIRCUITOS BASICOS CON SCR

Este módulo tiene los mismos elementos en hardware que el anterior módulo pero este no cuenta con control de ángulo remoto, por lo tanto no posee sensores especiales de corriente y tensión, tampoco cuenta con sistema de comunicaciones especiales. Este módulo cuenta con diodo volante y transformador reductor para implementar diferentes topologías de control de fase, solamente se tiene control de ángulo de disparo manualmente como se puede observar en la siguiente figura 39 y figura 40.



Figura 39. A. Modulo Circuitos Basicos con SCR.



Figura 40. B. Modulo Circuitos Basicos con SCR

### 3.3 Sistema de Alimentación del sistema

Se implementó una fuente de alimentación con las siguientes características:

Salidas de tensión DC: 12[v], -12[v] y 5[v].

Salida de corriente DC: 1[A].

Estos valores de tensión y corriente se obtuvieron a partir de la reducción de tensión y regulación de la señal de la red eléctrica como se muestra en la figura 42. Y figura 41. Los niveles obtenidos son suficientes para alimentar los sistemas de adquisición, control, comunicaciones y acoplamiento de las diferentes señales.

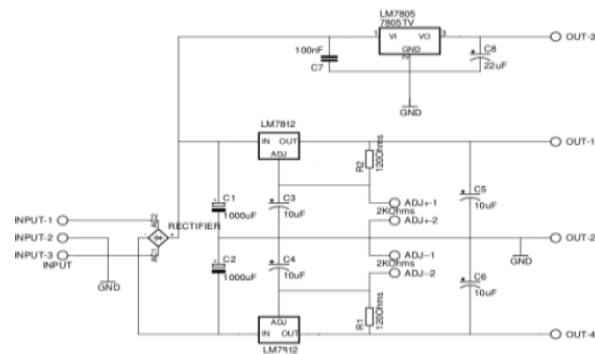


Figura 41. Esquema de circuitol.



Figura 42. Circuito realizado

### 3.4 Sistema de detección de cruce por cero

Se implementó con optoacopladores 4N25 que permiten mantener un aislamiento físico con la red eléctrica y no causar posibles daños al sistema microcontrolado

como se muestra en la figura 43. El cruce por cero se refiere a la detección del punto donde una señal seno cambia de semiciclo, ya sea de semiciclo positivo a semiciclo negativo o de semiciclo negativo a semiciclo positivo como se muestra en la figura 44,45 y 46, cuando la detección ocurre el sistema genera un impulso de duración corta pero suficiente para generar una tarea específica en el microcontrolador, tareas como controlar dispositivos (tiristores) para el control de potencia a una carga en forma sincronizada.

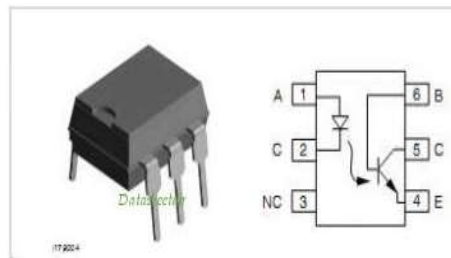


Figura 43. MOC 4N25

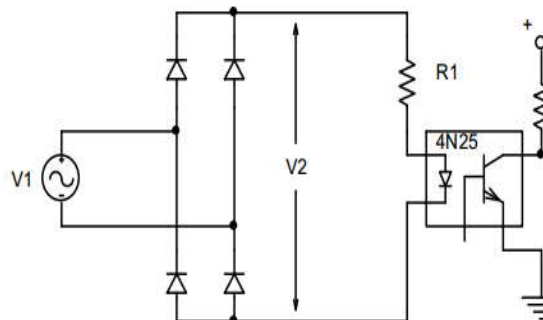


Figura 44. Detección de cruce por cero

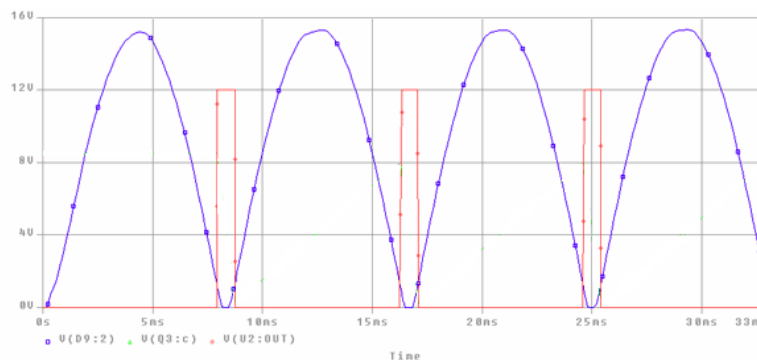


Figura 45. Señal obtenida

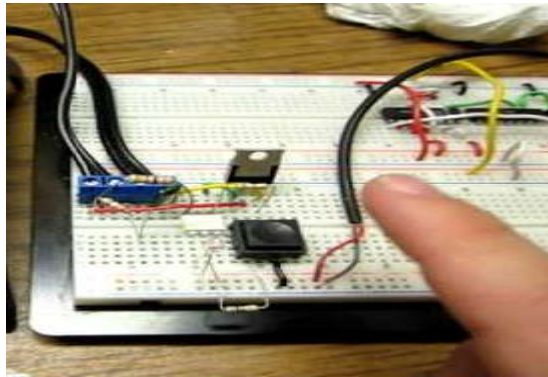


Figura 46. Montaje físico

### 3.5 Captura y Acondicionamiento de señal de tensión

La señal de tensión se adquirió a través de un divisor de tensión ubicado directamente en la alimentación a la carga, el divisor de tensión se calculó para que la muestra de la señal que tiene como máximo una amplitud de 120 voltios con una variación del 10% sea de 3vpp ver figura 47, la muestra alimenta un amplificador operacional de instrumentación de aislamiento denominado AD202JN que aísla la señal de la parte de potencia para su análisis ver figura 48, una vez obtenida la señal de baja potencia es sumada con un señal dc por medio de un amplificador operacional común como se muestra en la figura xxx, este procedimiento se realiza generalmente por que los microcontroladores solamente digitalizan señales de corriente continua (solamente semiciclos positivos).

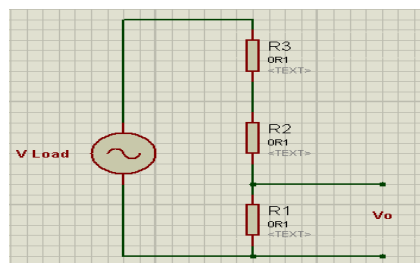


Figura 47. Divisor de tensión

El AD 202JN es un amplificador de propósito general de dos canales en este tipo de amplificadores no hay continuidad óhmica entre la entrada y la salida, presenta dos transformadores de aislamiento, el uno utilizado para aislar la señal y el otro para proveer una fuente de voltaje aislada de  $\pm 7.5$  voltios y 0.4 mA.

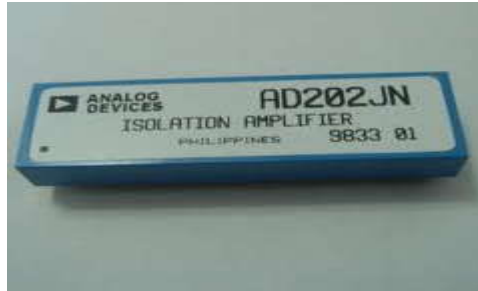


Figura 48. Amplificador de instrumentación AD202JN

Las tensiones máximas de aislamiento que soporta son:

- 1500 Vrms de voltaje AC, 60 Hz, continuos.
- $\pm 2000$  Vpico continuos (AC o DC)

Requiere una fuente de alimentación de 15 voltios, e internamente posee moduladores y demoduladores para convertir la señal en corriente alterna para reconstruirla las señales después ver figura 49.

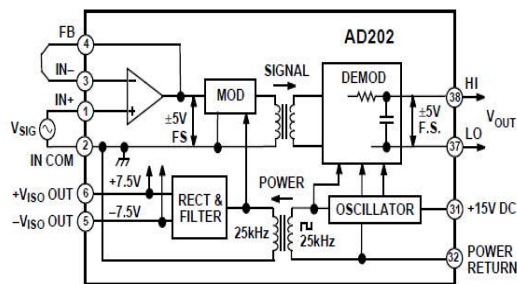


Figura 49. Diagrama de Bloques

Dónde:

- IN + : Entrada positiva del amplificador
- IN COM : Entrada / Común de fuente de voltaje aislada
- IN - : Entrada Negativa del amplificador
- FB : Entrada de realimentación
- HI : Salida LO
- LO : Salida HI
- - Viso out : Fuente de voltaje aislada -7.5
- + Viso out : Fuente de voltaje aislada +7.5
- POWER SUPPLY (15V.)
- POWER RETURN

Con el fin de evitar la distorsión de la señal se ha configurado el operacional de aislamiento como seguidor.

Como se mencionó anteriormente el acondicionamiento de la señal de tensión obtenida se ejecutó con un opam LM324 como se muestra en la figura 50.

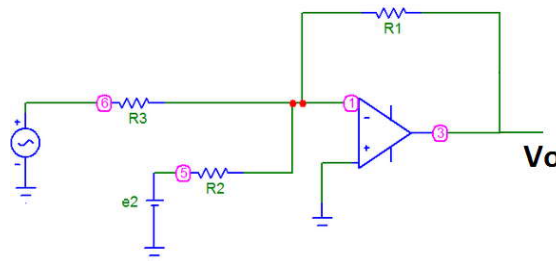


Figura 50. Circuito de acondicionar la señal para su digitalización



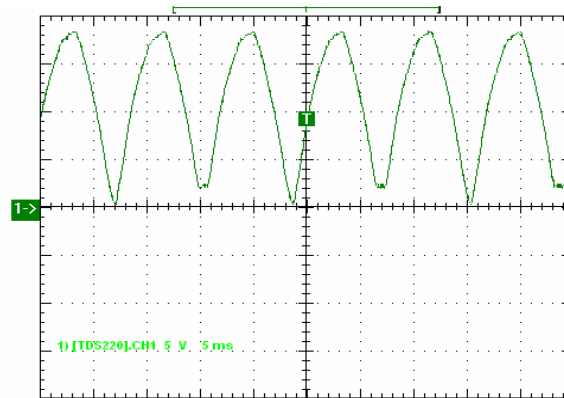


Figura 51. Señal obtenida

Como se puede observar la señal obtenida está por encima de la referencia y por lo tanto puede ser digitalizada por el microcontrolador.

### 3.6 Captura y Acondicionamiento de señal de Corriente

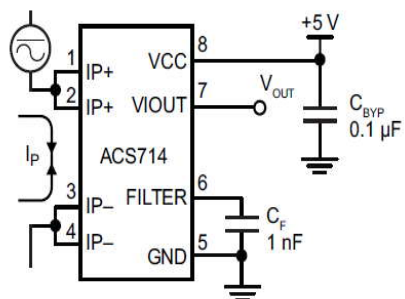


Figura 52. ACS714

La señal de corriente se adquirió a través de un sensor de efecto hall figura 52 figura 53, colocado en serie con la carga el cual se empotro en una tarjeta para su mejor manipulación, a continuación se presentan sus características:

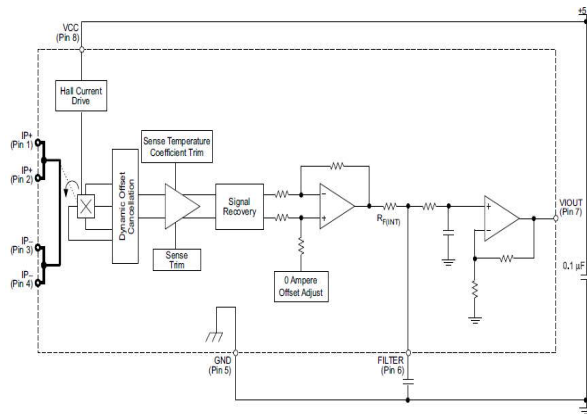


Figura 53. Diagrama interno del ASC714

Dónde:

Terminal List Table

Number	Name	Description
1 and 2	IP+	Terminals for current being sensed; fused internally
3 and 4	IP-	Terminals for current being sensed; fused internally
5	GND	Signal ground terminal
6	FILTER	Terminal for external capacitor that sets bandwidth
7	VIOUT	Analog output signal
8	VCC	Device power supply terminal

La tarjeta con sensor de corriente ACS714LLC-05B, tiene un rango de medición de -5 a 5 amperios, con sensibilidad de 185mV/A.

El IC tiene una resistencia para filtro en el pin #6, la cual con un condensador de 1nF, produce un filtro paso bajo con frecuencia de corte a 80 KHz para mejorar la precisión de sistema. Para aplicaciones de detección de baja frecuencia añadiendo un condensador en paralelo con el condensador de 1nF este se puede ubicar en la tarjeta como "Cf". La frecuencia de corte F a la que el filtro atenúa a la mitad de su potencia original está dada por:

$$= \frac{1}{2} = \frac{1}{10681 \times 1}$$

Donde Cf es el valor del capacitor a añadir en la tarjeta.

Finalmente el sistema se integra en una tarjeta impresa como se muestra en la siguiente figura 54.

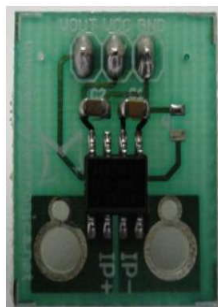


Figura 54. Circuito impreso ACS714

La tensión obtenida en el terminal Vout es proporcional a la corriente censada, como se muestra en la siguiente figura 55.

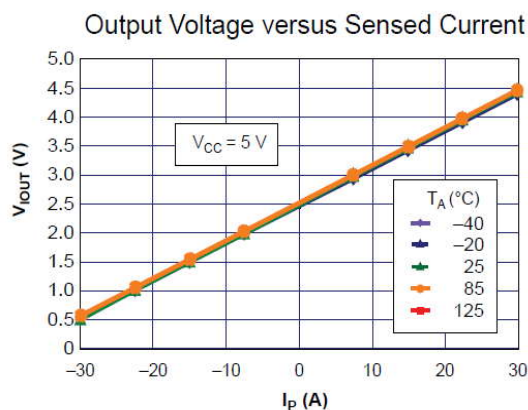


Figura 55. Tensión vs Corriente

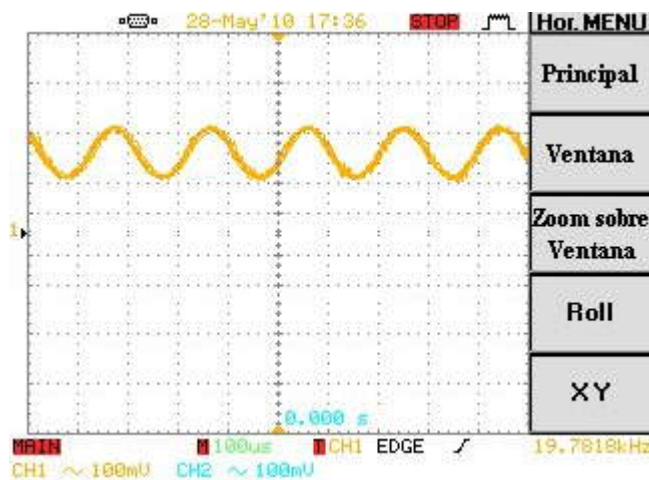


Figura 56. Señal obtenida del sensor de corriente

Como se pudo observar en la figura 56, anterior el resultado de la magnitud de la corriente capturada es siempre positivo, razón por la cual no necesita ser sumado a una señal dc para digitalizar la señal análoga, esta representación de la señal de corriente es pasada a través de un amplificador operacional en modo seguidor para prevenir posibles caídas de tensión debido a la impedancia de entrada del microcontrolador.

### 3.7 Temporizador LM555

La familia de circuitos integrados a la cual pertenece el LM555 es ampliamente conocida ya que los 555 en general son utilizados como temporizadores para aplicaciones tanto de retardo como de oscilación en equipo comercial, militar e industrial, ver figura 57.

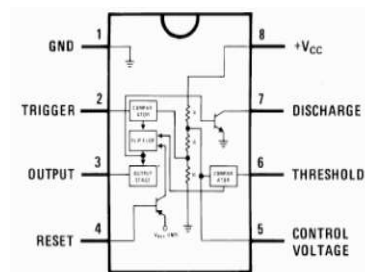


Figura 57. Estructura interna y diagrama de conexión del LM555

El LM555 es un temporizador altamente estable, ya que estos circuitos integrados monolíticos pueden producir retardos muy precisos. Entre sus características distintivas se tiene que pueden temporizar en ámbitos que van desde microsegundos hasta horas, haciendo que sea un circuito muy flexible para aplicaciones en las cuales se necesite que operen en modo astable y monoestable. Son dispositivos muy útiles en aplicaciones de osciladores astables ya que pueden mantener la frecuencia de manera muy precisa y su ciclo de trabajo es completamente ajustable por medio de dos resistencias externas y un capacitor. Los circuitos implementados con el LM555 se pueden resetear y disparar con el flanco decreciente de la señal, además la salida puede entregar o

absorber hasta 200 mA. puede manejar dispositivos TTL. Entre las aplicaciones más comunes de este circuito se encuentran los temporizadores de precisión, generadores de pulsos, temporizadores secuenciales, detectores de pulsos, generadores de retardo, moduladores de ancho de pulso y de posición, así como generadores de rampa lineal.

### 3.7.1 Operación en modo monoestable

El LM555 en modo monoestable genera una señal con ancho de pulso ajustable cada vez que se presente la señal de disparo.

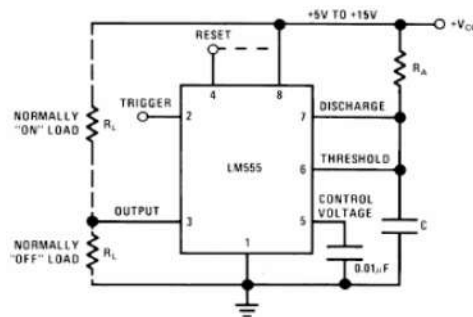


Figura 58. LM555 en configuración monoestable

El pulso monoestable se presenta en la salida (patilla número 3), su valor es  $V_{CC}$  y su duración se ajusta mediante la resistencia  $R_A$  y el capacitor  $C$ , de acuerdo con la siguiente ecuación.

$$T=1.1RAC.$$

El LM555 tiene un margen de error de aproximadamente 1% en la determinación de este tiempo. El ancho del pulso se inicia cada vez que el voltaje en la entrada de disparo (patilla número 2) llega a un valor menor a  $1/3 V_{CC}$ . Si durante la duración del pulso se aplica de nuevo un pulso en bajo en la entrada de disparo esto no provocará que el dispositivo se reinicie, pues solamente hasta la culminación del tiempo del pulso es cuando la salida vuelve a estar en estado bajo, y se debe tener muy claro que el solo en bajo se puede reactivar el pulso monoestable. Esta implementación es útil cuando se necesita generar un retardo como el del control por fase de los tiristores.

### 3.8 Sistema de comunicación RS232

La comunicación serial consiste en el envío de un bit de información de manera secuencial, esto es, un bit a la vez y a un ritmo acordado entre el emisor y el receptor.

La comunicación serial en computadores ha seguido los estándares definidos en 1969 por el RS-232 (Recommended Standard 232) que establece niveles de voltaje, velocidad de transmisión de los datos, etc. Por ejemplo, este protocolo establece un nivel de -12v como un uno lógico y un nivel de voltaje de +12v como un cero lógico (por su parte, los microcontroladores emplean por lo general 5v como un uno lógico y 0v como un cero lógico).

Existen en la actualidad diferentes ejemplos de puertos que comunican información de manera serial (un bit a la vez). El conocido como “puerto serial” ha sido gradualmente reemplazado por el puerto USB (Universal Serial Bus) que permite mayor versatilidad en la conexión de múltiples dispositivos. Aunque en naturaleza serial, no suele referenciarse de esta manera ya que sigue sus propios estándares y no los establecidos por el RS-232.

La mayoría de los microcontroladores, entre ellos Motorola Freescale, poseen un puerto de comunicación serial. Para comunicarse con los computadores personales actuales que poseen únicamente puerto USB requieren de un dispositivo “traductor”. Motorola Freescale emplea el integrado [FT232RL](#) figura 59, el cual es un convertidor USB-Serial. A través de este integrado el microcontrolador puede recibir y enviar datos a un computador de manera serial.



Figura 59. FT232

Este dispositivo se empotra en una tarjeta con socketMiniUSB para una implementación del hardware, como se muestra en la figura 60.

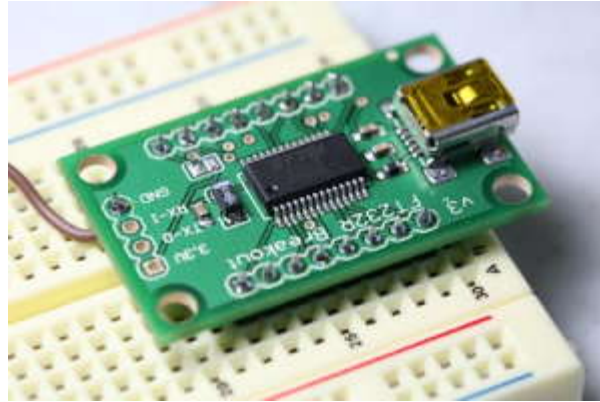


Figura 60. Tarjeta USB-Serial conFT232RL

La parte física encargada de la comunicación serial es la UART (Universal Asynchronous Receiver and Transmitter). Los microcontroladores Motorola Freescale, disponen de un dispositivo compatible llamado USART (Universal Synchronous and Asynchronous serial Receiver and Transmitter) que permite tanto la comunicación asincrónica como sincrónica figura 61 y figura 62.

En la comunicación asincrónica, la velocidad de envío de los datos es acordada a priori entre el emisor y el receptor. En la comunicación sincrónica, el envío de los datos es sincronizado por el emisor a partir de un pulso constante de reloj (Clock), con cada pulso envía un nuevo dato.

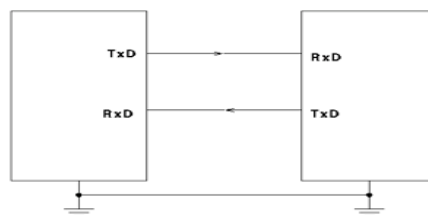


Figura 61.Comunicación asincrónica

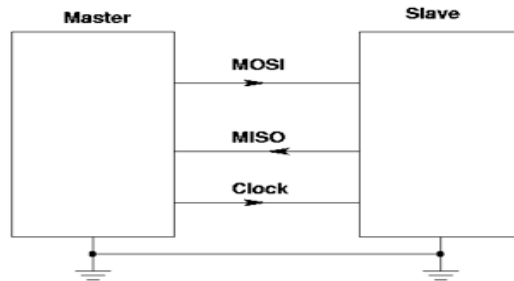


Figura 62. Comunicación sincrónica

### 3.8.1 Envío de datos

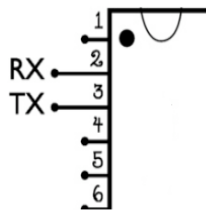


Figura 63. Pines de comunicación

En la comunicación con el computador Motorola-Freescale emplea la comunicación asincrónica. Esto es, requiere de sólo dos líneas de conexión que corresponden con los pines (Rx) y (Tx) pin de transmisión, y del establecimiento de un nivel de tierra común con el computador, esto es, ambas tierras deben estar conectadas, estableciendo el mismo nivel de voltaje de referencia. Figura 63.

Además de realizar las conexiones físicas entre el microcontrolador y el computador, para que pueda establecerse la comunicación serial debe existir un acuerdo previo en la manera cómo van a ser enviados los datos. Este acuerdo debe incluir los niveles de voltaje que serán usados, el tamaño y formato de cada uno de los mensajes (número de bits que constituirán el tamaño de la palabra, existirá o no un bit de inicio y/o de parada, se empleará o no un bit de paridad), el tipo de lógica empleada (qué voltaje representará un cero o un uno), el orden en



que serán enviados los datos (será enviado primero el bit de mayor peso o el de menor peso) y la velocidad de envío de datos.

### **3.9 Software del módulo no controlado, semicontrolado, controlado.**

En software este módulo cuenta dos diseños principales, el primer diseño se refiere a la implementación de un protocolo de comunicación serial hecho en Python que se convierte por hardware en protocolo USB, y el procesamiento, conectividad en red y toma de decisiones hecho en Java en los ordenadores Cliente - Servidor, el segundo diseño se refiere al procesamiento, captura de señales, comunicación serial y toma de decisiones en los microcontroladores hecho en lenguaje Ensamblador y lenguaje de programación C para microcontroladores.

#### **3.9.1 Implementación protocolo comunicación serial a 256000 bps en PYTHON**

Como se mencionó anteriormente este protocolo se implementó en Python por que permite la manipulación del set point de velocidades de comunicación que no son estándares, debido a que la transmisión es asincrónica, es posible enviar datos por un línea mientras se reciben datos por otra. Las características más importantes de la comunicación serial son la velocidad de transmisión (baudrate) que se configuro 256000 bps. Bits de datos que son la cantidad de bits en la transmisión que se colocaron de en un paquete de 8 bits. Bit de parada usado para indicar el fin de la comunicación. Paridad es una forma sencilla de verificar si hay errores en la transmisión serial. Existen cuatro tipos de paridad: par, impar, marcada y espaciada. La opción de no usar paridad alguna también está disponible.

Este lenguaje es segmentado y además la sintaxis de Python es muy parecida al Matlab, la captura de información se realiza en 6 paquetes de 8, configurados de la siguiente forma, 2 bytes de cabecera de la trama, 2 bytes de la muestra de la señal de corriente y, 2 bytes de la muestra de la señal de cómo se muestra a continuación:

```
import socket
import threading
import serial
import array
```

```
class SerialManager(threading.Thread):
```

```
    def __init__(self, port, out):
        threading.Thread.__init__(self)
        port = port.replace("\n", "")
        self.serial = serial.Serial(port, 256000)
        self.serial.flush()
        self.serial.flushInput()
        self.serial.flushOutput()
        self.out = out
        self.__run = True
        self.__errcnt = 0
```

```
    def stop(self, stopSerial=True):
```

```
        self.__run = False
        if stopSerial:
            self.serial.close()
        if self.out != None:
            self.out.close()
```

```
    def run(self):
```

```
        while self.__run:
            if self.out != None:
                try:
                    val = self.serial.read()
                    if val != "":
                        self.out.send(str(ord(val)) + "\n")
                except Exception as w:
                    print "Error SerialManager:"
                    print w
```

```
self.__errcnt += 1
if self.__errcnt==5:
    self.stop()
```

```
def procesar(cadena, _socket, parent):
    if cadena.startswith("iniciar:"):
        h = SerialManager(cadena.split(":")[1], _socket)
        h.start()
        return h
    elif cadena.startswith("detener"):
        parent.hilo.serial.close()
        parent.hilo.stop()
    elif cadena.startswith("enviar"):
        val = int(cadena.split(":")[1].replace("\n", ""))
        try:
            pass
            parent.hilo.serial.write("\xff")
            parent.hilo.serial.flushOutput()
            print "==>", val
            parent.hilo.serial.write(array.array('B', [val]).tostring())
        except Exception as w:
            print "Error Procesamiento:"
            print w
    return parent.hilo
```

```
class ProcessThread(threading.Thread):
```

```
    def __init__(self, cli):
        threading.Thread.__init__(self)
        self.__run = True
        self.__cli = cli
        self.__errcnt = 0
        self.hilo = None
```

```
    def stop(self):
```

```
self.__run = False
if self.__cli!=None:
    self.__cli.close()
if self.hilo!=None:
    self.__cli.close()
```

```
def run(self):
    while self.__run:
        try:
            c = -1
            res = ""
            while c != "\n":
                c = self.__cli.recv(1)
                res += c
            self.hilo = procesar(res,self.__cli,self)
        except Exception as w:
            print "Error ProcessThread:"
            print w
            self.__errcnt += 1
            if self.__errcnt==5:
                self.stop()
```

```
class Receiver(threading.Thread)
    def __init__(self):
        threading.Thread.__init__(self)
        self.__server = socket.socket()
        self.__server.bind(("0.0.0.0",1982))
        self.__server.listen(1)
        self.__run = True
```

```
def stop(self):
    self.__run = False
def run(self):
    while self.__run:
```

```
self.__cli, data = self.__server.accept()  
ProcessThread(self.__cli).start()
```

```
r = Receiver()  
r.start()  
# SerialManager("COM12",None).start()
```

Implementación de Procesamiento, conectividad (Cliente-Servidor) y toma de decisiones en JAVA

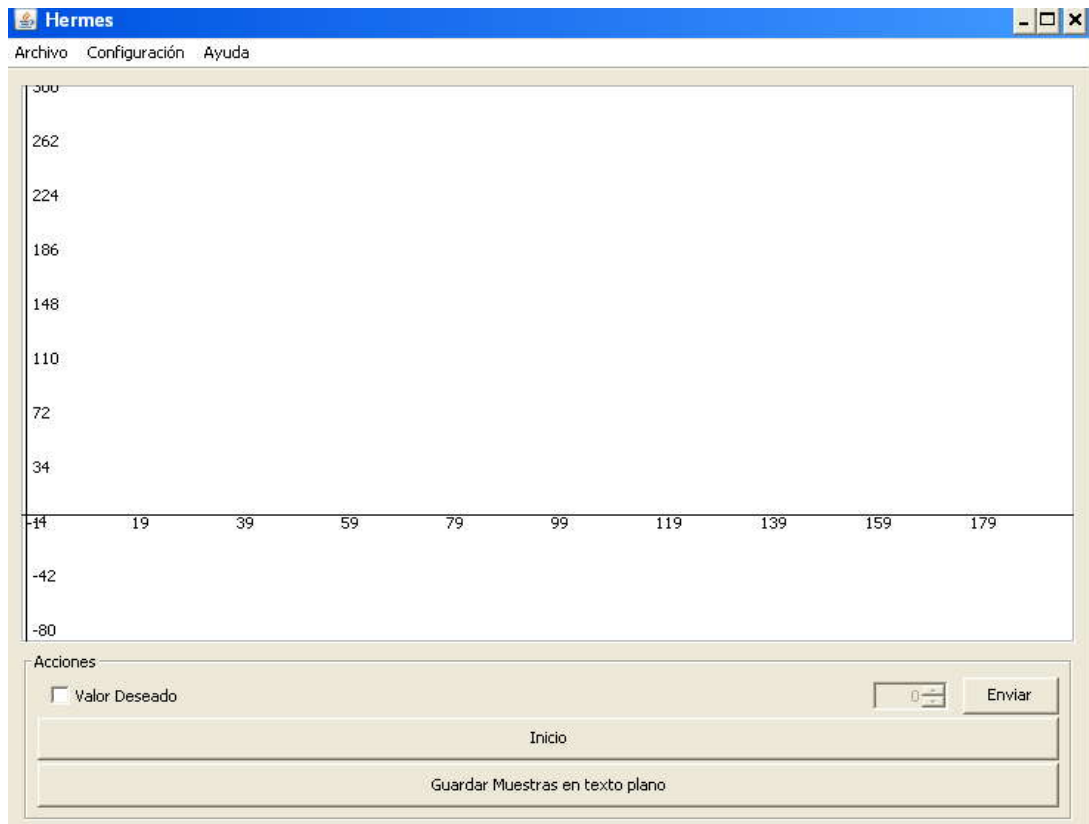


Figura 64. Ventana del Aplicativo

Se debe ingresar el valor de referencia o el valor deseado de la tensión promedio o  $V_{dc}$  luego se presiona enviar y el sistema de control define el ángulo de disparo.

Una vez obtenida la información proveniente de Python los 6 paquetes son procesados de la siguiente forma:

Se implementó un buffer de 6 bytes, Los dos primeros paquetes de 8 bits son el encabezado de la trama, indicando que llegó al puerto un dato, estos paquetes siempre tienen un valor fijo y es el máximo valor que puede ocupar byte, El resto de la información tiene una característica especial, la información de tensión y corriente llegan en subpaquetes de 2 bytes esto es debido al conversor analógico digital implementado en el microcontrolador, por lo tanto se debe realizar una operación matemática de producto y suma con los subpaquetes para obtener el valor real de tensión y corriente. Una vez obtenidos estos valores se guardan en un buffer para visualizar en pantalla del servidor y del cliente las señales de corriente y tensión.

El buffer obtenido nos sirve para la implementación de estrategias de control del sistema de potencia porque nos permite tener muestras en tiempo real, donde generalmente el resultado de la estrategia implementada es la toma de decisiones con respecto al ángulo de disparo de los tiristores que regulan la energía a la carga.

La visualización en pantalla se obtuvo imprimiendo el buffer de almacenamiento cuando este está lleno y así continuamente lo que se realiza alrededor de 10ms.

A continuación se muestra la estructura de captura, procesamiento, conectividad cliente-Servidor, visualización que se realizó en Java, ver figura 65.

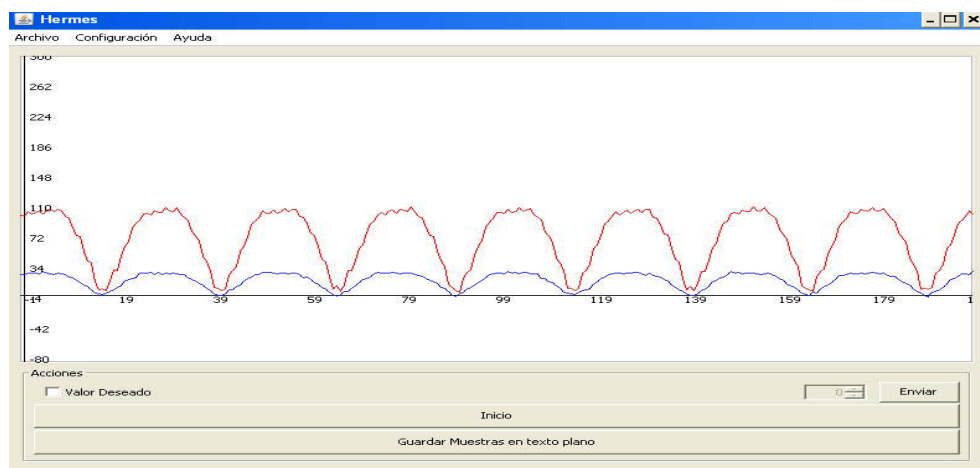


Figura 65. Señales obtenidas de tensión y corriente

Para la comunicación cliente servidor se crearon las siguientes clases

- Cliente.java
- Mensaje.java
- Serial.java
- SerialComunication.java
- Servidor.java

Los datos obtenidos del muestreo de la señal se guardan en una clase denominada

- Datos.java

Se construyeron dos hilos

- Hilolectura.java
- InterfazHiloVista.java

La integración de clases y entorno de trabajo se realizó en la carpeta de ejecución hermes

Los códigos detallados se entregan en el anexo A.

### **3.10 Implementación de la comunicación, captura de señales y el procesamiento en los microcontroladores Motorola Freescale.**

#### **3.10.1 Comunicación:**

Se creó para enviar información entre uno de los microcontroladores y el computador host, este protocolo se desarrolló porque este microcontrolador no posee en sus registros la velocidad específica requerida para enviar los datos de tensión y corriente al pc entre muestra y muestra, y aunque estos dispositivos poseen velocidades de conexión más altas, no se tomaron en cuenta ya que se podía presentar traslape de información. A pesar que los microcontroladores se pueden programar en lenguajes de alto nivel se decidió implementar el software

de transmisión (TXD) y recepción (RXD) en lenguaje ensamblador porque este lenguaje permite tener más control desde la raíz del sistema en términos de ciclos de máquina. El protocolo de comunicación desarrollado usa dos pines de propósito general y no utiliza los que tiene embebido el sistema, los pines pertenecen al puerto B, el software en ensamblador se muestra a continuación:

Transmisión:

TXD:

```
bclr 2,PTBD ;transmision BIT DE START
sthx ciclos10 ; 10/2 ciclos
sthx ciclos10 ; 10/2 ciclos
sthx ciclos10 ; 10/2 ciclos
ldhx #$0009
clc
lda ten1
```

rotar:

```
lsra
decx
bnesaca_bit_dato
jmp fin
```

saca\_bit\_dato:

```
bcc saca_0
bset 2,PTBD
sthx ciclos10 ;
sthx ciclos10 ;
sthx ciclos10 ;
sthx ciclos10 ; 10/2 ciclos
bra rotar
```

saca\_0:

```
bclr 2,PTBD
nop
sthx ciclos10 ;
sthx ciclos10 ;
sthx ciclos10 ;
sthx ciclos10 ; 10/2 ciclos
bra rotar
```

fin:

```
bset 2,PTBD ;// transmision BIT DE STOP
sthx ciclos10 ;
sthx ciclos10 ;
sthx ciclos10 ;
sthx ciclos10 ; 10/2 ciclos
sthx ciclos10 ; 10/2 ciclos
```



rts ; fin de sbrutina de envio de datos

Recepción:

```
RXD:
lda  #$00
sta  kbipe      ;pin interrupción desactivado
clc
```

esperar:

```
brset 1,ptbd,esperar ;inicio espera de dato real
clrx
jsr  nada
jsr  nada
jsr  nada
nop
nop
nop
nop
nop
nop
```

solo10:

```
brset 1,ptbd,in1 ;llega un uno '1'
nop
nop
nop
nop
nop
clc      ;0 en carry
ldx  dato
rorx
stx  dato
lda  cont
inca
sta  cont
cmpa #8
bne solo10
```

bra termino

nada: rts

Estos códigos se ejecutaron a 125 nS por ciclo de máquina y a una tasa de 3 ciclos por instrucción.

### 3.10.2 Captura de señales Análogas y digitales:

#### 3.10.2.1 Captura de señales Análogas:

Se efectuó uso del módulo especial de conversión Análogo - digital con el que cuenta el microcontrolador, el cual entrega como resultado de la conversión una representación binaria de 10 bits de la señal análoga que percibe el pin x del puerto A. esta representación binaria se transfiere a dos registros de 8 bits de propósito general declarados en el software, por lo tanto la información de corriente y tensión es enviada al computador "host" en 16 bits. La tensión referencia que se utilizó para la adquisición es la de alimentación positiva, A continuación se puede observar el código para realizar esta tarea:

ADC:

```
        lda    #$00      ;tensión
        sta    adcsc1
no_v:   brclr  7,ADCSC1,no_v
        lda    ADCRH
        sta    vhigh
        lda    ADCRL
        sta    vlow
        nop
        nop
        LDA    #$01      ;corriente
        STA    ADCSC1
no_i:   BRCLR  7,ADCSC1,no_i
        lda    ADCRH
        staihigh
        lda    ADCRL
        stailow
        nop
        nop
        lda    #255
        sta    ten1
        jsr    TXD
        lda    #255
```

```

sta  ten1
jsr  TXD
lda  vhigh
sta  ten1
jsr  TXD
lda  vlow
sta  ten1
jsr  TXD
lda  ihigh
sta  ten1
jsr  TXD
lda  ilow
sta  ten1
jsr  TXD
lda  #$0A

```

sigo:

```

deca
cmpa  #00
bne  sigo
feed_watchdog
bra   mainLoop

```

### 3.10.2.2. Captura de señales Digitales:

Estas señales son las que provienen del detector de cruce por cero y que son periódicas cada 120 Hertz como se muestra en la figura 66, esto significa que a esta frecuencia se tiene un periodo de 8.3333 ms,

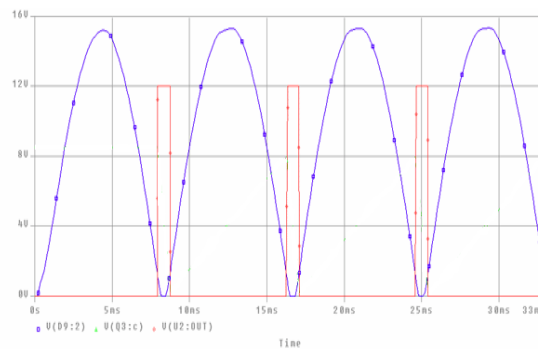


Figura 66. Señal Digital Cruce por cero

Esta señal es capturada por el microcontrolador por el pin de interrupción IRQ que es un pin digital pues solo acepta dos cambios de estado alto o bajo, esta interrupción indica al controlador el cero para la ejecución de control de ángulo de disparo, los grados para los disparos se convierten en tiempo si se sabe que la frecuencia de la red eléctrica es de 60 Hertz, con esta información se conoce el periodo de la señal que será aproximadamente de 16.6666ms, y que el periodo de una señal es de 360 grados, por lo tanto se puede conocer en términos del tiempo la magnitud de un grado a 60 Hertz, esta conversión se realiza por medio de una regla de tres simple como se muestra:

$$180^{\circ} = 8.3333\text{ms}$$

$$1^{\circ} = x [\text{ms}] \quad x = 46.2962\text{us}$$

El microcontrolador es un dispositivo que trabaja ejecutando ciclos de maquina por lo tanto debemos conocer cuántos ciclos se requieren para contar un grado, si cada ciclo de maquina en este dispositivo se ejecuta a 125ns tendremos:

$$\text{Numero de ciclos} = 46.2962 \text{ uS} / 125 \text{ nS} = 371 \text{ ciclos aproximadamente.}$$

Esta cantidad se ciclos que corresponden a un grado que se almacena en una variable de valor fijo ya que esta magnitud no variará.

El valor del ángulo de disparo se puede obtener mediante dos formas:

A - Manualmente

B - Por Control (comunicación con el pc)

A – Manualmente: Se obtiene por medio de una rutina de interrupción del microcontrolador, la cual se ejecuta cuando una de las entradas que se configuraron por KBD (Key board interrupt ) cambia de estado alto a bajo, este cambio es generado por una pulsador que se configuro como pull-up en el módulo.

B – Por Control: el ángulo obtenido es el resultado de un método matemático de control donde intervienen un modelo de la planta, valores de tensión y corriente, y un valor de referencia deseado.

Una vez adquirido el ángulo de disparo en términos de ciclos de maquina el valor es asignado a un registro del módulo de temporización TPMMOD, este módulo TPM compara el valor ingresado con el valor del registro de conteo TPMCONT cuando se alcanza el valor deseado se genera una interrupción que ejecuta una subrutina donde un pulso de magnitud 1ms de ciclo útil es creado para el disparo de los tiristores

El pulso es ingresado a un Im555 configurado como oscilador monoastable que invierte la señal para tener un mejor acoplamiento de impedancias entre los sistemas. Luego la señal ingresa a un driver Hex inverting Schmitt trigger que excita los optoacopladores que son los encargados al final de dar el disparo al SCR como se muestra en las figura 67 y figura68.

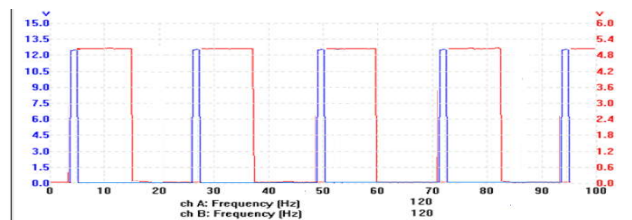


Figura 67. Excitación a los optoacopladores señal de entrada (rojo), señal de salida (azul).

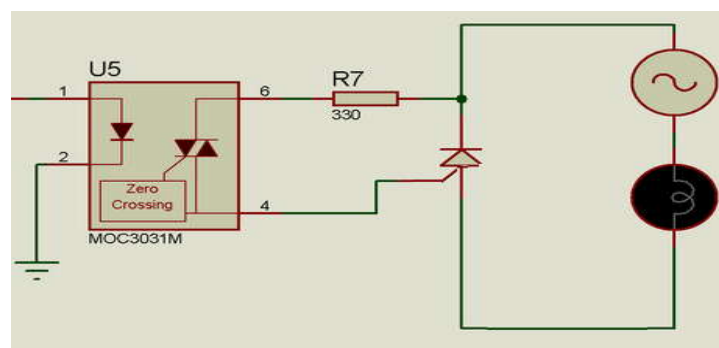


Figura 68. Circuito de excitación al tiristor

## Capítulo 4. Validación de los módulos

Estos resultados provienen de implementar varias topologías de tipo resistivo o con carga RL, en los dos módulos que se implementaron.

### 4.1 Control PI al Convertidor Controlado

En esta etapa se parte de la referencia que será ingresada por el usuario al sistema de control para dar una respuesta al requerimiento.

El modelo que se asemeja a la planta en este sistema es de primer orden por que la carga al sistema es resistiva. A continuación se muestra en la figura 69, la planta sin controlador.

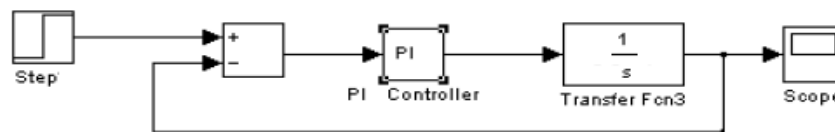


Figura 69. Controlador PI

Se analiza el modelo PI de control, en forma análoga hasta llegar a los parámetros PI para un control digital.

$$u(t) = kp \left( e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau \right)$$

$$U(s) = kp \left( E(s) + \frac{1}{T_i s} E(s) + T_d s E(s) \right)$$

$$\frac{U(s)}{E(s)} = kp \left( 1 + \frac{1}{T_i s} \right)$$

$$\frac{U(s)}{E(s)} = \left( Kp + \frac{Ki}{s} \right)$$

Donde  $K_i = K_p / T_i$ .

$$\frac{de(t)}{dt} \approx \frac{e(kT) - e(kT - T)}{T} \quad \int_0^t e(t)dt \approx \sum_{k=1}^n T e(kT).$$

$$\langle V_o \rangle = \frac{1}{\pi} \int_{\alpha}^{\pi} V_p \sin(\omega t) d(\omega t) = \frac{V_p}{\pi} \cos(\alpha) + 1$$

$V_o$  También puede expresarse como

$$V_o = (1/T) [(\sum \text{Simple}) T / 180]$$

$$V_o = (\sum V_{\text{sample}}) / 180$$

$V_{\text{deseado}} = \text{Ref.}$ , es un valor que entra por consola, el sistema responde de la siguiente forma al sistema de control:

$$E_k = \text{Ref} - V_o$$

$$U = K_p E_k + (K_i \int E_k dt) = K_p E_k + (K_i U_i)$$

$$U_i = U_{i-1} + T_{\text{control}} / 2 (E_k - E_{k-1})$$

Esto se realiza en forma iterativa.

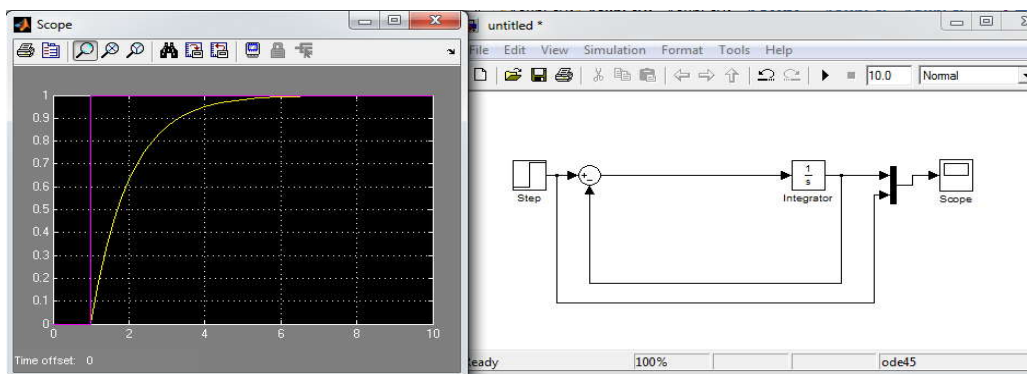


Figura 70. Planta con excitación escalón

Sistema con control proporcional después de experimentar con la ganancia hasta el límite de inestabilidad

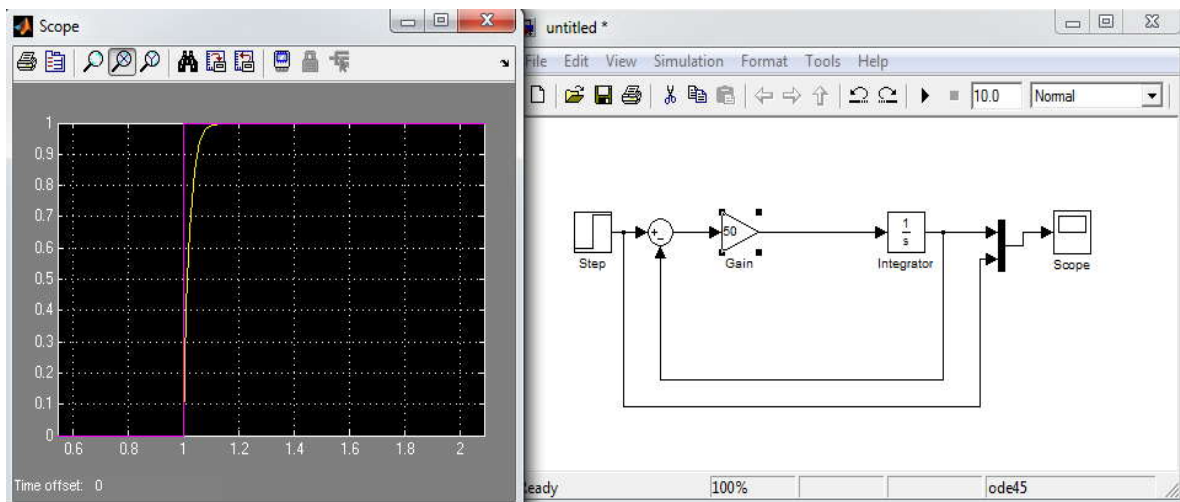


Figura 71. Planta Control Proporcional

Sistema con control Proporcional Integral hallada  $K_p$ .

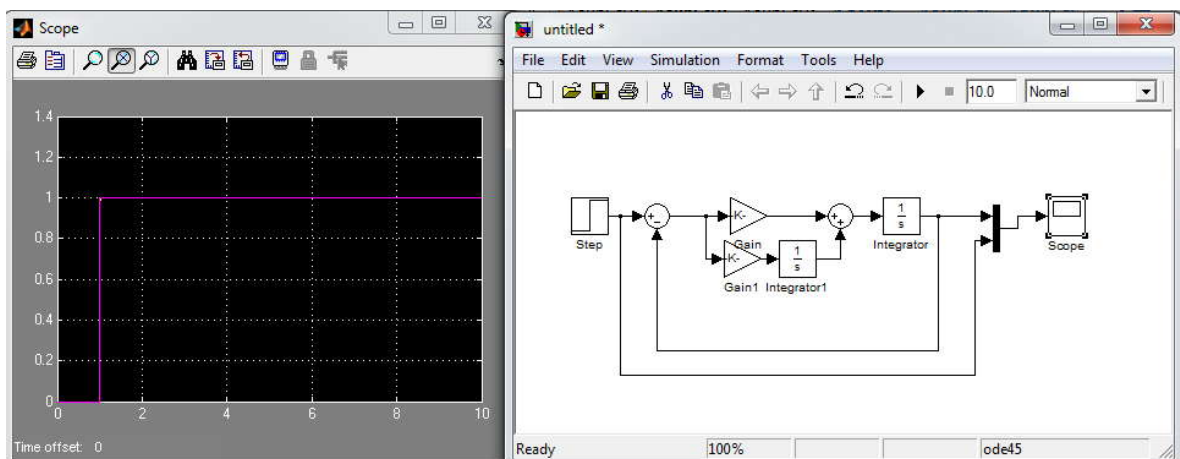


Figura 72. Planta con control Proporcional Integral



## 4.2 Modulo Control de Fase

Control de medio ciclo con carga  $R$ : se puede apreciar la forma de onda de salida del rectificador.

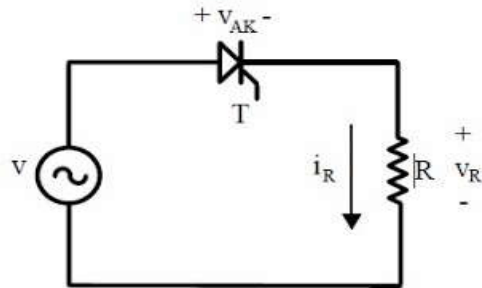


Figura 73. Control de un semiciclo carga  $R$

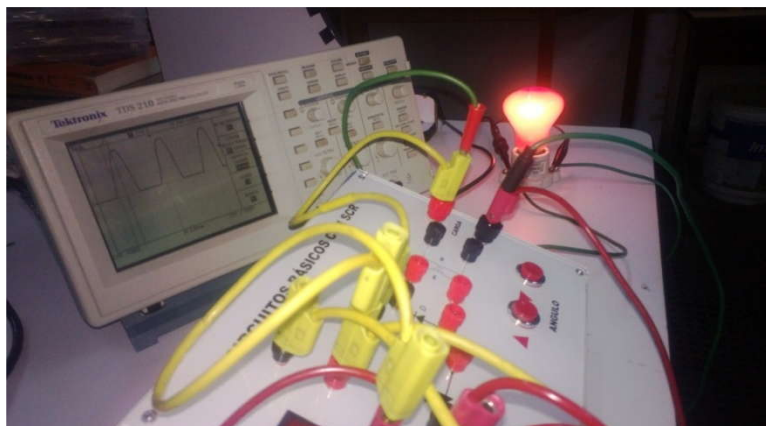


Figura 74. Control a  $0^\circ$  de carga  $R$ .



Figura 75. Control a  $90^\circ$  de carga  $R$ .

Control de medio ciclo con carga  $R - L$ : se puede apreciar la forma de onda de salida del rectificador.

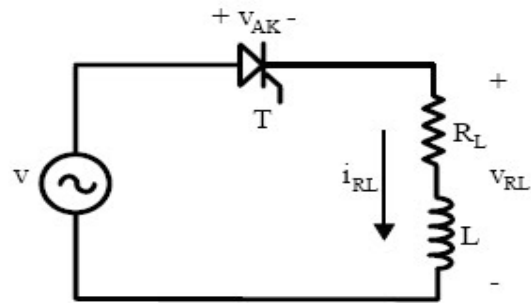


Figura 76. Control de un semiciclo con carga  $R - L$

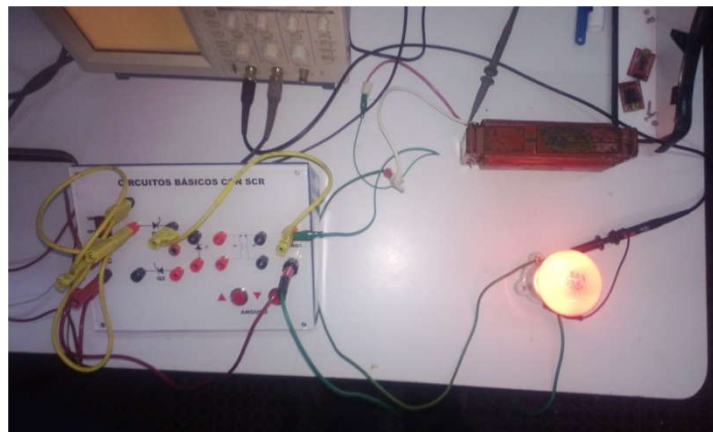


Figura 77. Montaje Físico

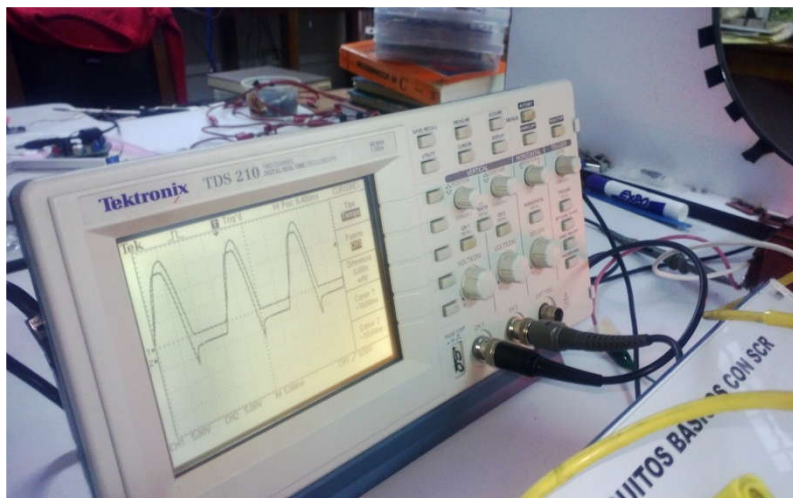


Figura 78. Control a  $0^\circ$  de carga  $R - L$

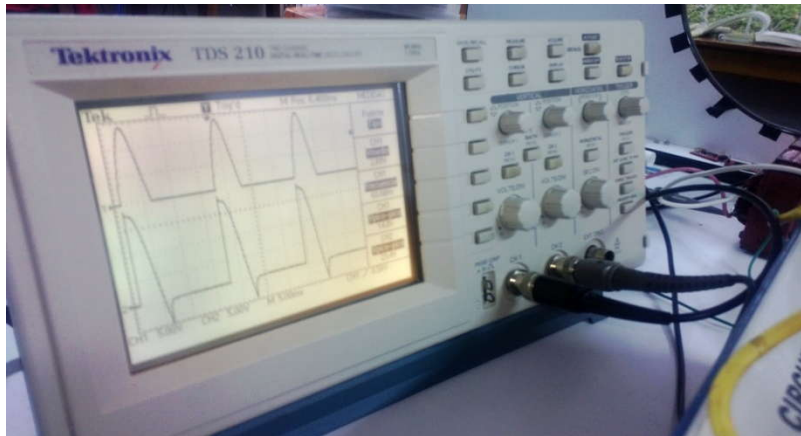


Figura 79. Control a  $90^\circ$  de carga R – L

Control de ciclo completo con carga  $R$ : se puede apreciar la forma de onda de salida del rectificador.

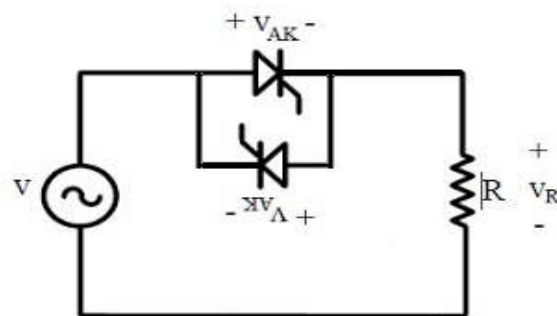


Figura 80. Control de fase

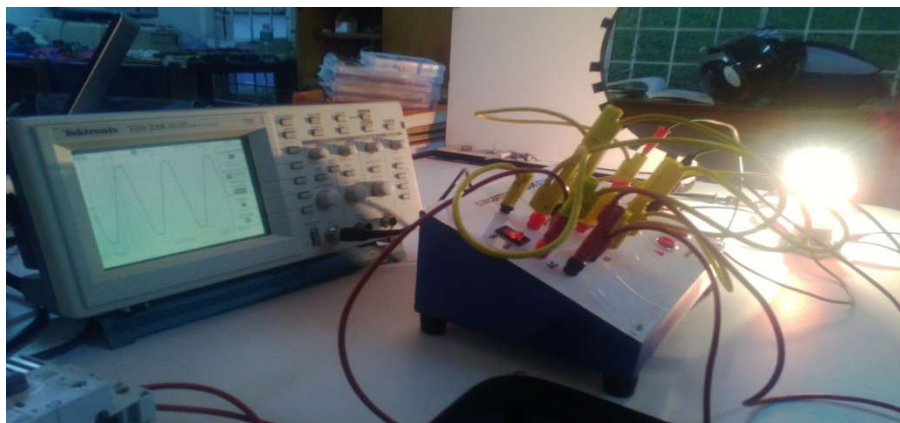


Figura 81. Control a  $0^\circ$  de carga  $R$

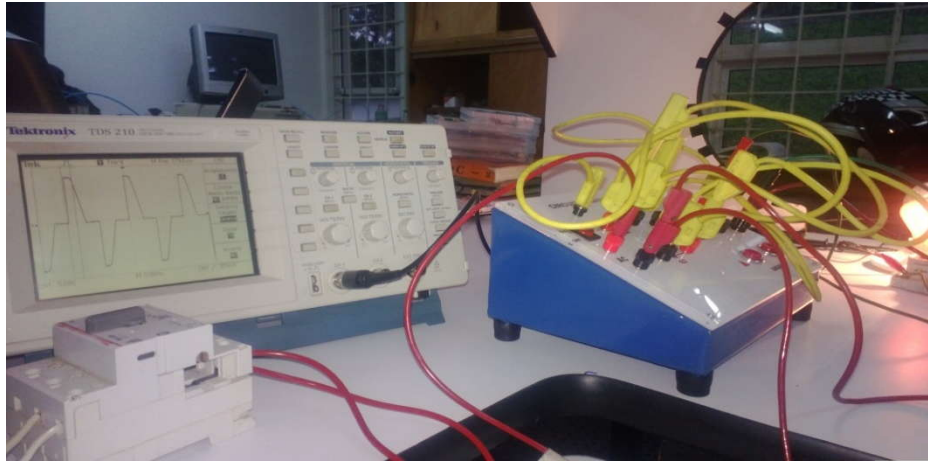


Figura 82. Control a  $90^\circ$  de carga R

Control de ciclo completo con carga  $R - L$ : se puede apreciar la forma de onda de salida del rectificador.

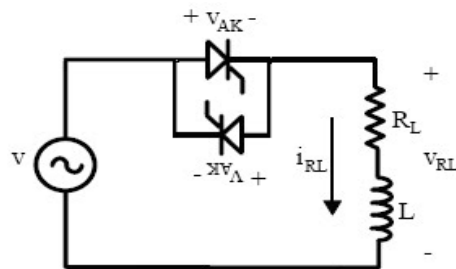


Figura 83. Control de fase con carga  $R - L$

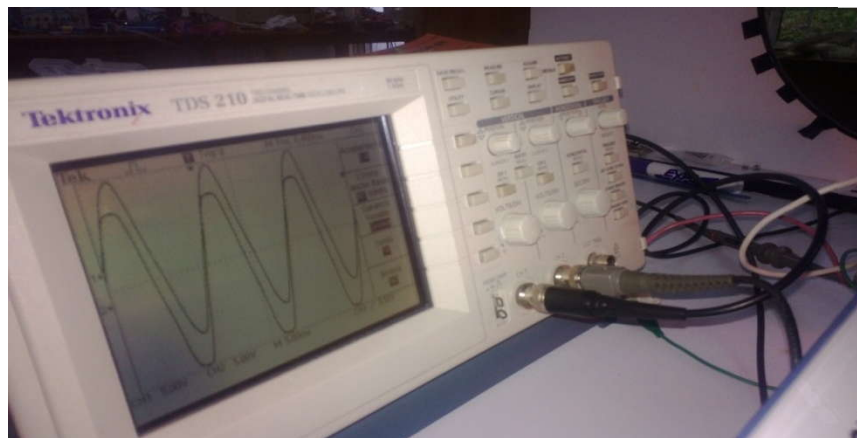


Figura 84. Control a  $0^\circ$  de carga  $R - L$

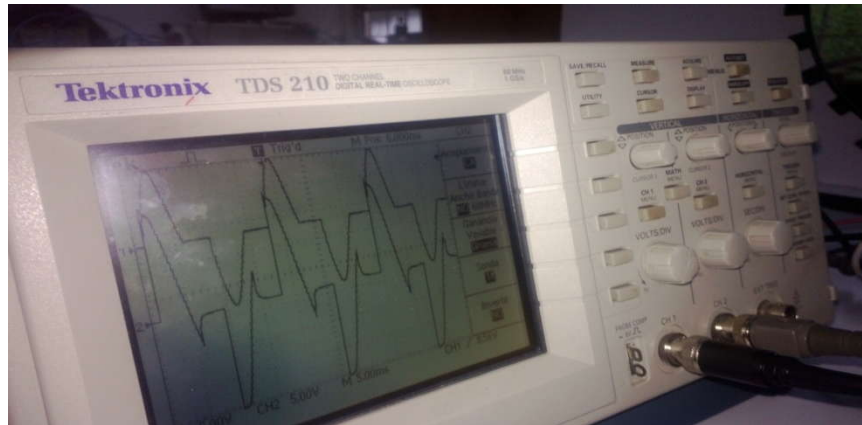


Figura 85. Control a 90° de carga R - L

### 4.3 Modulo Convertidor Controlado, Semicontrolado y no Controlado

Convertidor AC/DC No Controlado CON carga R

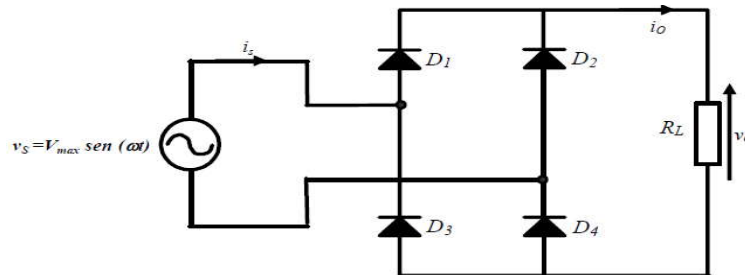


Figura 86. Rectificador no controlado con carga R

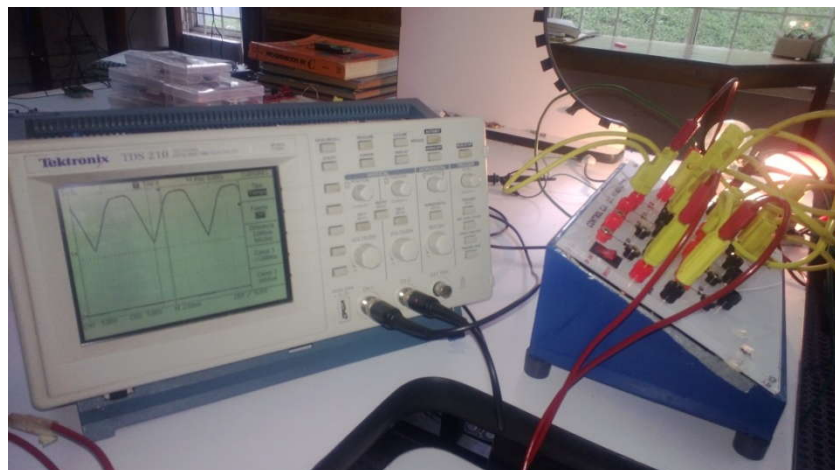


Figura 87. Convertidor a 0° carga R.



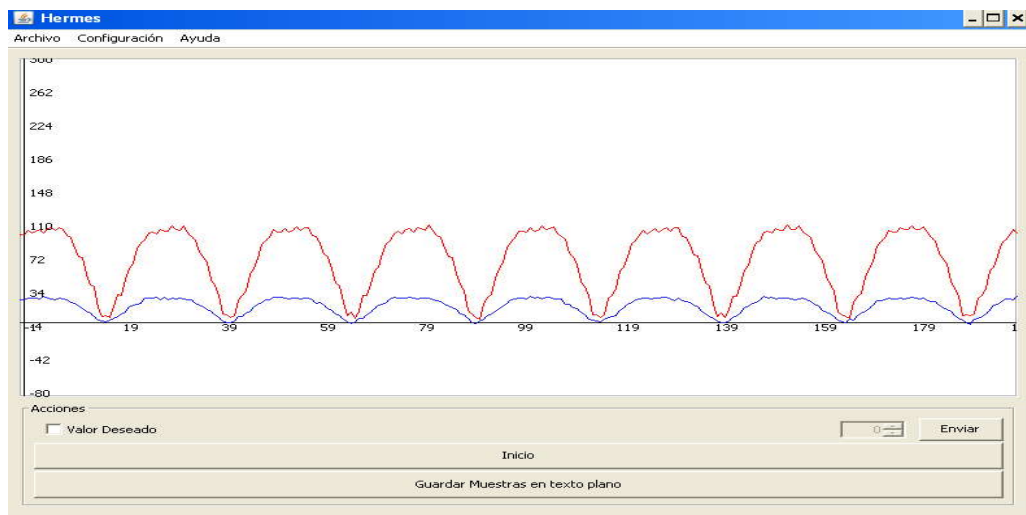


Figura 88. Convertidor a  $0^\circ$  carga R visto desde consola



Figura 89. Convertidor a  $90^\circ$  carga R.

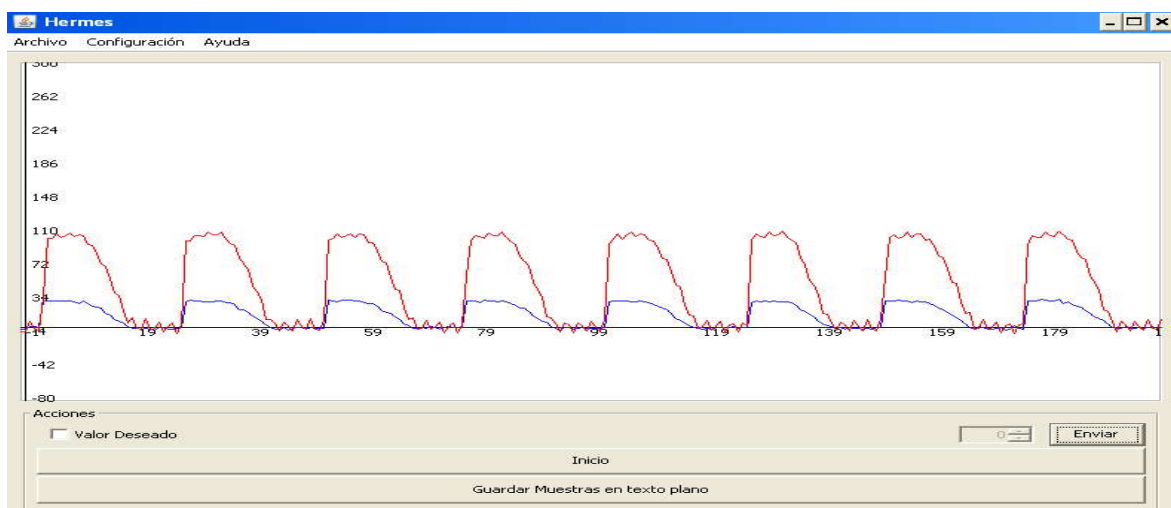


Figura 90. Convertidor a  $90^\circ$  carga R visto desde la consola

## Convertidor AC/DC No Controlado CON carga R – L

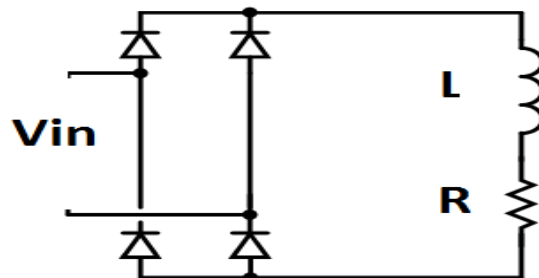


Figura 91. Convertidor AC/DC No Controlado CON carga R – L

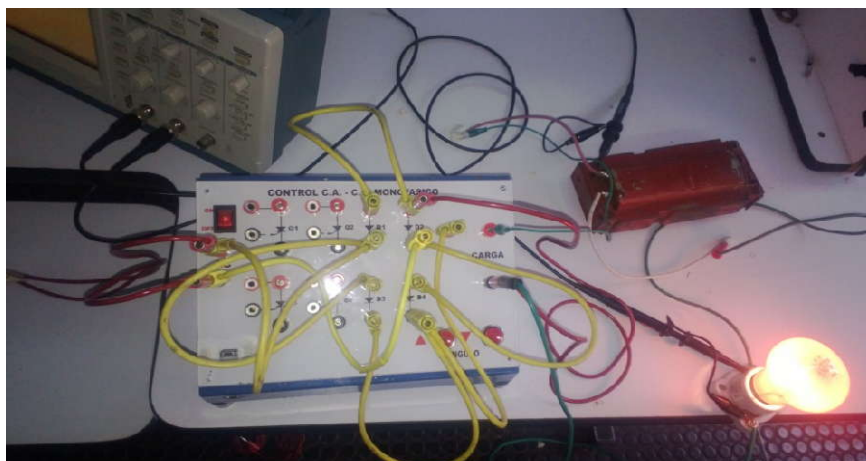


Figura 92. Montaje físico

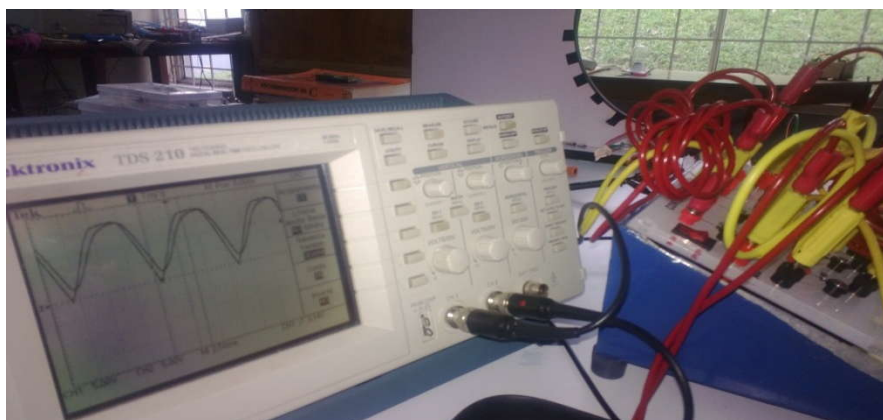


Figura 93. Control a  $0^\circ$  con carga R - L

## Convertidor AC/DC Semi-Controlado CON carga R

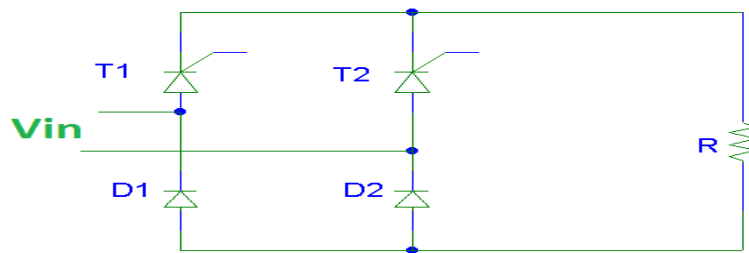


Figura 94. Convertidor AC/DC Semi-Controlado CON carga R

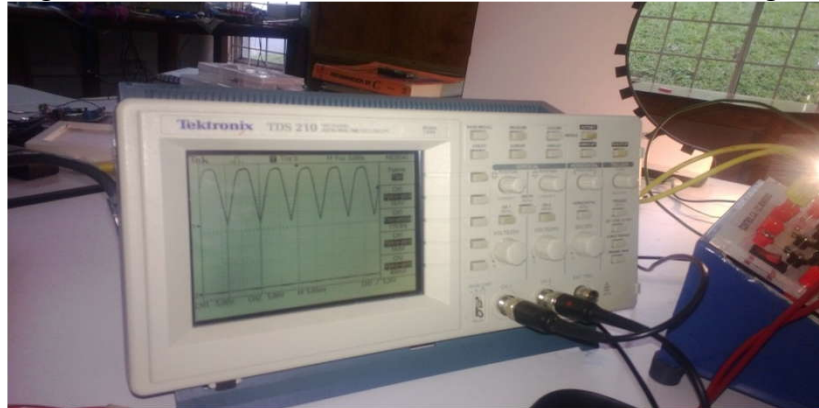


Figura 95. Control a  $0^\circ$  con carga R

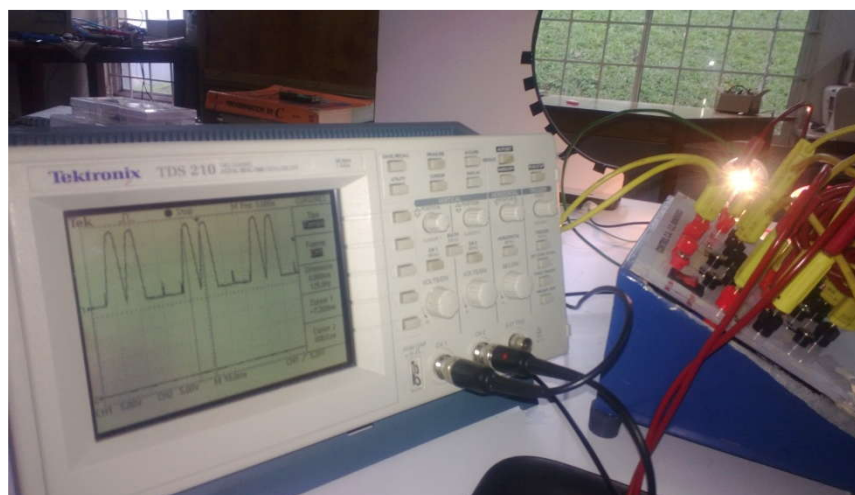


Figura 96. Control a  $90^\circ$  con carga R



## Convertidor AC/DC Semi-Controlado CON carga R – L

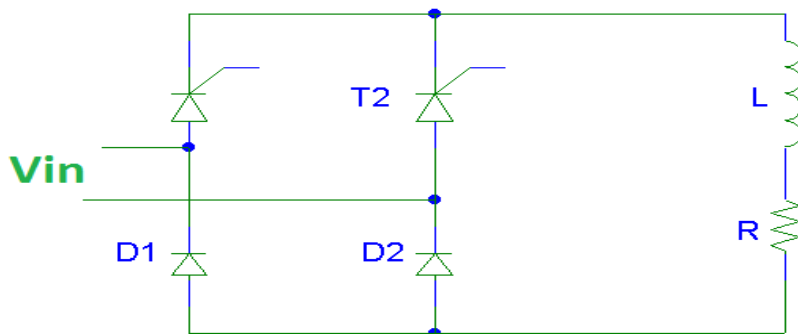


Figura 97. Convertidor AC/DC Semicontrolado CON carga R – L

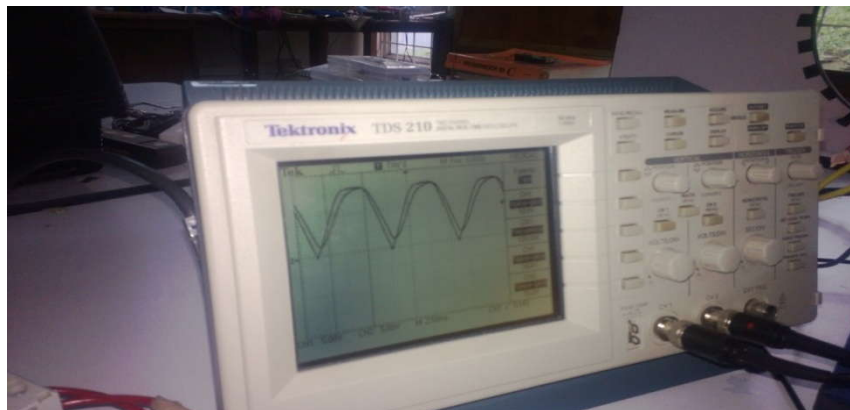


Figura 98. Control a  $0^\circ$  con carga R – L

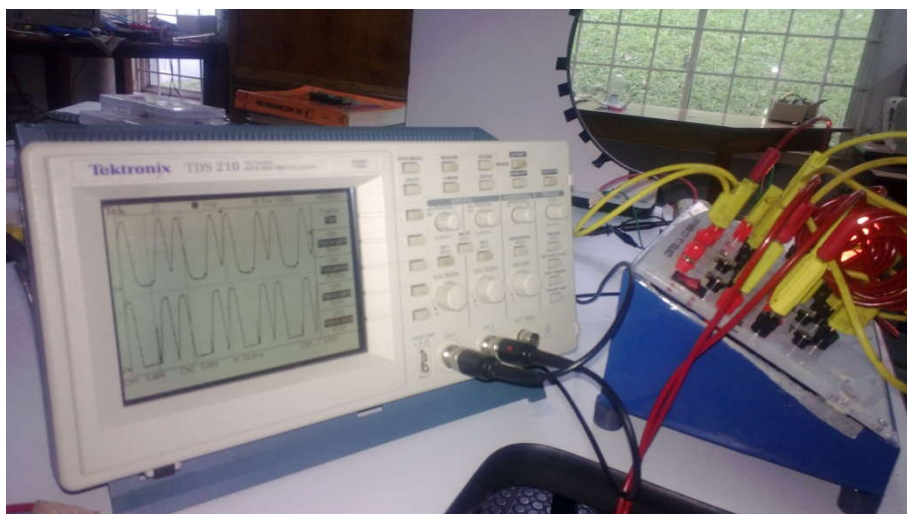


Figura 99. Control a  $90^\circ$  con carga R – L

## Convertidor AC/DC Controlado CON carga R

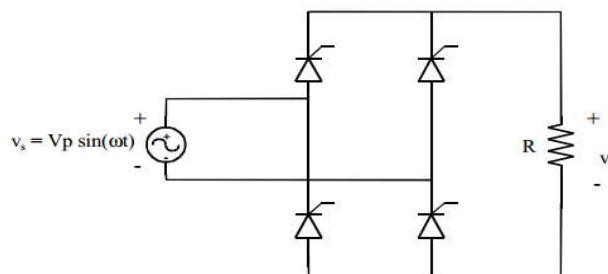


Figura 100. Convertidor AC/DC Controlado CON carga R

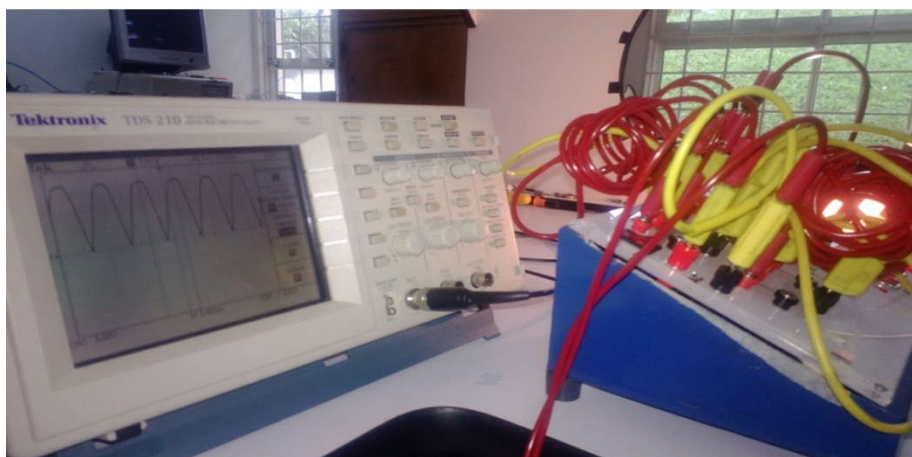


Figura 101. Control a  $0^\circ$  con carga R



Figura 102. Control a  $90^\circ$  con carga R

## Convertidor AC/DC Controlado CON carga R – L

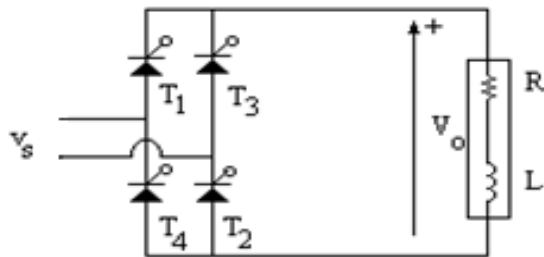


Figura 103. Convertidor AC/DC Controlado CON carga R – L

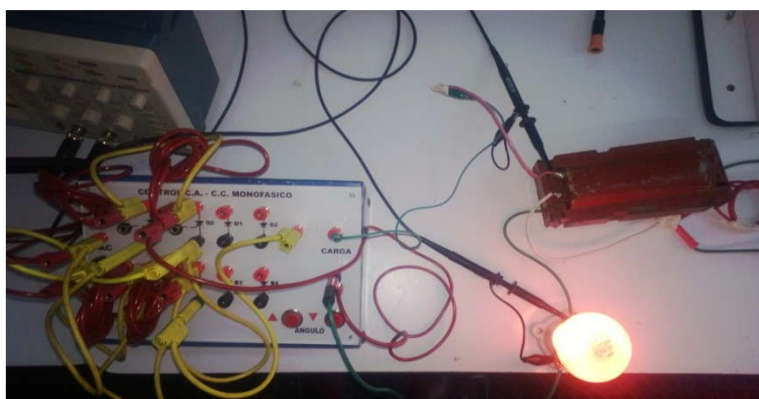


Figura 104. Montaje físico

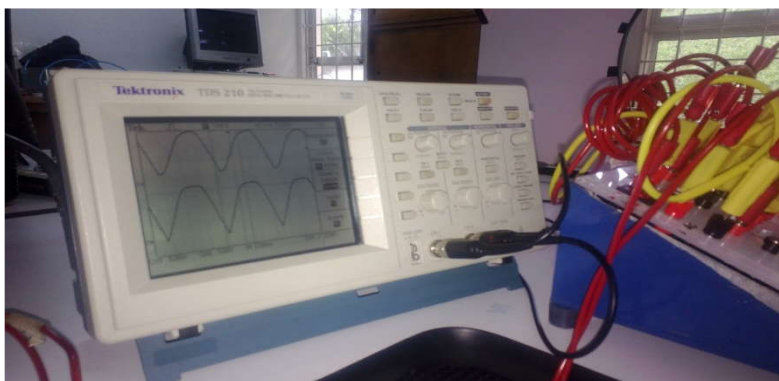


Figura 105. a. Control a  $0^\circ$  con carga R – L

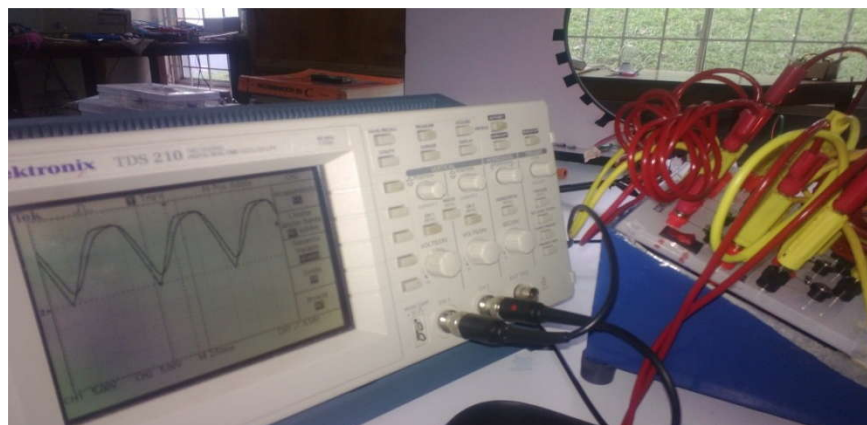


Figura 106. b. Control a  $0^\circ$  con carga R - L



Figura 107. Control a  $90^\circ$  con carga R - L

## Conclusiones

Durante el desarrollo del circuito se logró la realización de un diseño que efectivamente cumple con el requisito para el estudio de los parámetros necesarios en un rectificador controlado y semicontrolado de onda completa por medio de un control PI desde una ubicación remota, esto se puede observar en el hecho que se logró un circuito cuyo control se ejerce sobre una única variable  $R$ , la cual tiene una relación lineal con el parámetro de control y con una precisión experimental de alrededor de un 1%. Los demás parámetros del circuito se especificaron detalladamente de manera que se cumpliera con los requisitos deseados tomando en cuenta el ambiente y los recursos disponibles en para los laboratorios, de manera que el laboratorio no se concentre en problemas de especificación si no en el estudio y análisis del circuito diseñado.

La validación de resultados permitió cuantificar el desempeño del circuito real con respecto a los datos teóricos determinados mediante modelos matemáticos y computacionales, encontrando resultados muy satisfactorios de manera que se espera que si la guía se sigue correctamente no se presente ningún problema que entorpezca el entendimiento del circuito y asegurando que bajo condiciones de laboratorio el estudiante podrá observar resultados valiosos para análisis con un montaje remoto..

## Bibliografía

1. Timothy L. Skvarenina *"The power electronics handbook"*, CRC Press, Boca Raton London New York Washington, D.C, 2002
2. Robert W. Erickson, Dragan Maksimović *"Fundamentals of power electronics"*, Segunda edición, Kluwer Academic Publishers, New York, Boston Dordrecht, London, Moscow, 2004.
3. Muhammad H. Rashid, *"Power Electronics Handbook"*, Academic Press, San Diego, San Francisco, New York, Boston, London. Sydney, Tokyo, 2001
4. Ramos Ramos, Guillermo, Hernández M., Jorge Eduardo y Castaño Welgos, Juan Andrés. *"Curso Práctico De Electrónica Industrial Y Automatización"* Cedit, 2002.
5. Jaycar, *"Optocouplers: When & How To Use Them"*  
[http://www.jaycar.com.au/images\\_uploaded/optocoup.pdf](http://www.jaycar.com.au/images_uploaded/optocoup.pdf)
6. Tejada P. Alexis, Llamas T. Armando, *"Efectos De Las Armónicas En Los Sistemas Eléctricos"*  
<http://www.mty.itesm.mx/etie/deptos/ie/profesores/allamas/cursos/ueee/armonicas/07Efectarm.PDF>
7. a b What's New in Python 2.0
8. PEP 227 -- Statically Nested Scopes
9. PEPs 252 and 253: Type and Class Changes
10. PEP 255: Simple Generators
11. PEP 282 -- A Logging System
12. threading — Higher-level threading interface

13. "Holandés" hace referencia a Guido van Rossum, el autor del lenguaje de programación Python, que es holandés. También hace referencia a la gran concentración de desarrolladores holandeses conocidos en relación a otras nacionalidades.
14. <http://docs.python.org/tutorial/controlflow.html#defining-functions>
15. Flanagan, David (2002). *JavaScript: The Definitive Guide* (4<sup>a</sup> Edición edición). [ISBN 0-596-00048-0](#).
16. [Douglas Crockford on Functional JavaScript](#) (2:49): "[JavaScript] es el lenguaje funcional más popular del mundo. JavaScript es y siempre ha sido, al menos desde [la versión] 1.2, un lenguaje de programación funcional."

